



HAL
open science

Proposition d'une méthodologie de réingénierie pour le contrôle par le produit de systèmes manufacturiers : Application au circuit du médicament d'un hôpital

Jean-Charles Jch Huet

► To cite this version:

Jean-Charles Jch Huet. Proposition d'une méthodologie de réingénierie pour le contrôle par le produit de systèmes manufacturiers : Application au circuit du médicament d'un hôpital. Sciences de l'ingénieur [physics]. Université Blaise Pascal - Clermont-Ferrand II, 2011. Français. NNT: . tel-00636342v1

HAL Id: tel-00636342

<https://theses.hal.science/tel-00636342v1>

Submitted on 27 Oct 2011 (v1), last revised 19 Dec 2011 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : D.U. : 2099
EDSPIC : 513

Université BLAISE PASCAL - Clermont II
École Doctorale Sciences Pour l'Ingénieur de Clermont-Ferrand

THÈSE

présentée pour obtenir le grade de Docteur d'Université,
spécialité « Informatique »

par

Jean-Charles HUET

PROPOSITION D'UNE MÉTHODOLOGIE DE RÉINGÉNIERIE POUR LE CONTRÔLE PAR LE PRODUIT DE SYSTÈMES MANUFACTURIERS :

Application au circuit du médicament d'un hôpital

Thèse soutenue le 07/01/2011 devant le jury composé de :

M.	XAVIER BOUCHER	Professeur	(Examineur)
M ^{me}	CORINE CAUVET	Professeur	(Rapporteur)
M.	JEAN CHOPINEAU	Professeur	(Examineur)
M.	MICHEL GOURGAND	Professeur	(Directeur)
M.	ALAIN GUINET	Professeur	(Examineur)
M.	KHALID KOUISS	Maître de Conférences	(Co-encadrant)
M.	JEAN-LUC PARIS	Professeur	(Directeur)
M.	ANDRÉ THOMAS	Professeur	(Rapporteur)
M.	VINCENT PINON	Pharmacien praticien hospitalier	(invité)

Réalisé en collaboration entre le Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes (LIMOS) et le Centre Hospitalier Universitaire (CHU) de Clermont-Ferrand

À mes proches...

*« L'homme et sa sécurité doivent constituer la première préoccupation de toute aventure
technologique. »*

Albert Einstein

REMERCIEMENTS

Les travaux de recherche présentés dans ce mémoire ont été développés au Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes de Clermont-Ferrand (LIMOS, UMR 6168 du CNRS) dirigé par le Professeur Alain QUILLOT que je remercie vivement pour son accueil au sein du laboratoire.

Un grand merci à Jean CHOPINEAU, Professeur d'Université et Chef du service Pharmacie au CHU de Clermont-Ferrand, pour m'avoir accueilli dans son service pendant ces trois années de thèse. Je remercie aussi l'ensemble du personnel de la pharmacie et tout particulièrement Vincent PINON, dont l'écoute ouverte et critique, l'expérience concrète du terrain, m'ont permis de réaliser un travail pragmatique. Merci aussi à eux pour avoir accepté d'être dans mon jury de thèse en tant qu'examinateur et invité. Un merci particulier à mes collègues de bureau : Jean-François et Michel. J'exprime également ma reconnaissance à l'équipe de l'atelier de modélisation du CHU et notamment à Monsieur Bernard ALEKSI, Virginie, Sophie et toutes les personnes que j'ai pu y croiser.

Au terme de ces travaux, je tiens tout particulièrement à remercier Madame Corine CAUVET, Professeur à l'Université Paul Cézanne (Aix-Marseille III), et Monsieur André THOMAS, Professeur à l'Université Henri Poincaré (Nancy I), pour avoir accepté d'évaluer cette contribution scientifique en qualité de rapporteurs. Je remercie également Monsieur Xavier BOUCHER, Professeur à l'École Nationale Supérieure des Mines (ENSM) de Saint-Etienne, ainsi que Monsieur Alain GUINET, Professeur à l'Institut National des Sciences Appliquées (INSA) de Lyon, qui a accepté d'être président du jury, pour l'intérêt qu'ils ont porté à ces travaux en tant qu'examineurs.

Je tiens à exprimer ma reconnaissance et mes remerciements les plus sincères à mes directeurs de thèse, Monsieur Michel GOURGAND, Professeur à l'Université Blaise Pascal (Clermont II), et Monsieur Jean-Luc PARIS, Professeur à l'Institut Français de Mécanique Avancée (IFMA), ainsi qu'à mon co-encadrant Khalid KOUISS, Maître de Conférences à l'IFMA, pour avoir dirigé mon travail. Un merci particulier pour la qualité de leur collaboration, leurs nombreux conseils, leur aide constante et pour la façon efficace avec laquelle ils ont suivi ce travail.

Je remercie l'ensemble des membres du LIMOS, ainsi que le personnel du CHU, pour les fructueuses discussions que j'ai pu avoir avec eux, pour leurs conseils, pour leur soutien, et qui ont contribué, d'une manière ou d'une autre, à l'aboutissement de ces travaux. Je remercie aussi tous les stagiaires avec qui j'ai travaillé et qui ont permis de faire avancer cette thèse : Laure, Didier et Julien. Un merci particulier à Claire qui a relu ma thèse pour corriger les fautes d'orthographe.

Enfin, je remercie tous ceux qui m'ont accompagné au cours de toutes ces années, famille et amis, qui m'ont entouré tous les jours de leur soutien.

TABLE DES MATIÈRES

TABLE DES MATIÈRES	vi
LISTE DES FIGURES	x
LISTE DES TABLEAUX	xiii
LISTE DES ACRONYMES	xv
INTRODUCTION GÉNÉRALE	1
1 CONTEXTE ET PROBLÉMATIQUES	5
1.1 INTRODUCTION	6
1.2 LE CIRCUIT DU MÉDICAMENT DANS UN HÔPITAL	6
1.2.1 Fonctionnement du circuit du médicament	7
1.2.2 Objectif : la dispensation nominative	8
1.3 SYNCHRONISATION DES FLUX PHYSIQUES ET INFORMATIONNELS	12
1.3.1 Synchronisation des flux du CDM	12
1.3.2 Interopérabilité avec le Système d'Information Hospitalier	12
1.3.3 Interaction avec le flux physique	12
1.4 ANALOGIE ENTRE LE CDM ET UN FMS	14
1.5 CONCLUSION	17
2 LE CONTRÔLE PAR LE PRODUIT DES SYSTÈMES DE PRODUCTION	19
2.1 INTRODUCTION	21
2.2 LES DIFFÉRENTES APPROCHES DE PILOTAGE	21
2.2.1 Approche hiérarchique	21
2.2.2 Approche hétérarchique	22
2.2.3 Approche hybride	23
2.3 LE CONCEPT D'HOLON ET DE SYSTÈME MANUFACTURIER HOLONIQUE	24
2.3.1 PROSA	25
2.3.2 Autres architectures de HMS	26
2.3.3 Différence entre une approche basée sur les agents et une approche holonique	27
2.4 LE BESOIN DE PRENDRE EN COMPTE L'INTEROPÉRABILITÉ DANS UNE ENTREPRISE	28
2.4.1 Les différents niveaux de décision dans une entreprise	28
2.4.2 La notion de Système de Systèmes et le Système d'Information Hospitalier	31
2.4.3 Les modèles d'interopérabilité	32
2.4.3.1 Le modèle LISI	33
2.4.3.2 Le modèle LCIM	34
2.5 CONCLUSION	35

3	ÉTUDE DES OUTILS DE MODÉLISATION RELATIVES AUX SYSTÈMES DE PRODUCTION	37
3.1	INTRODUCTION	39
3.2	LES ARCHITECTURES DE RÉFÉRENCE	40
3.2.1	Le cadre de Zachman	40
3.2.2	CIMOSA	41
3.2.3	ARIS	42
3.2.4	L'ingénierie dirigée par les modèles (IDM) et l'architecture dirigée par les modèles (MDA)	43
3.3	LES MÉTHODOLOGIES DE MODÉLISATION	44
3.3.1	PERA	45
3.3.2	La méthodologie ASCI	45
3.3.3	Méthodologie Bidirectionnelle de Conception des Systèmes Automatisés (MBCSA)	47
3.3.4	Les méthodologies de conception des Systèmes Muti-Agents	48
3.3.4.1	MaSE	48
3.3.4.2	TROPOS	49
3.3.4.3	GAIA	50
3.4	LES MÉTHODES DE MODÉLISATION	51
3.4.1	IDEF-SADT	51
3.4.2	GRAI-GIM	53
3.4.3	Merise	53
3.4.4	L'environnement MEGA	53
3.5	LES LANGAGES DE MODÉLISATION	54
3.5.1	IEC 61499 - Les Blocs Fonctions	55
3.5.2	Le langage UML et ses extensions	56
3.5.2.1	AUML	56
3.5.2.2	UEML	56
3.5.2.3	SysML	57
3.5.2.4	UML pour le Temps Réel	57
3.5.3	LOTOS	57
3.5.4	L'analyse de déroulement, la norme JIS Z 8206	58
3.5.5	Formalisme pour l'analyse d'un flux de production	58
3.5.6	Les réseaux de Petri	61
3.6	UTILISATION DES MODÈLES	61
3.7	CONCLUSION	61
4	DE LA MAÎTRISE D'UN SYSTÈME HOLONIQUE	63
4.1	INTRODUCTION	64
4.2	MODÉLISATION D'UN SYSTÈME HOLONIQUE	64
4.3	MAÎTRISE D'UN SYSTÈME HOLONIQUE VIA LE CONTRÔLE PAR LE PRODUIT	67
4.4	CONCLUSION	70
5	NOTRE APPROCHE MÉTHODOLOGIQUE : RÉINGÉNIERIE POUR LE CONTRÔLE PAR LE PRODUIT DE SYSTÈMES MANUFACTURIERS	71
5.1	INTRODUCTION	73
5.2	CHOIX DES OUTILS DE MODÉLISATION	73
5.2.1	Choix de l'approche	73
5.2.2	Choix de l'architecture	74

5.2.3	Choix du langage	76
5.3	VOCABULAIRE EMPLOYÉ	77
5.3.1	Mise au point sur le terme « générique »	77
5.3.2	Transformation entre niveaux d'abstraction	77
5.4	LE NIVEAU M ₃ : DESCRIPTION D'UN SYSTÈME HOLONIQUE DE PRODUCTION	78
5.5	LE NIVEAU M ₂ : MODÉLISATION DU HMS	80
5.5.1	Représentation d'un HMS contrôlé par les ressources	81
5.5.2	Représentation d'un HMS contrôlé par le produit	81
5.5.3	Transformation pour la réingénierie	82
5.6	LIEN ENTRE LES TROIS DERNIERS NIVEAUX	82
5.6.1	MBCSA	82
5.6.2	ASCI	83
5.6.3	Bilan	85
5.7	MÉTHODOLOGIE DE RÉINGÉNIERIE DES HMS	85
5.7.1	Une méthodologie composée de deux parties	85
5.7.2	Utilisation à partir du méta-modèle	89
5.7.3	Utilisation à partir du MGC d'un domaine	90
5.8	CONCLUSION	92
6	INSTANCIATION ET APPLICATION AU DOMAINE DU CIRCUIT DU MÉDICAMENT	95
6.1	INTRODUCTION	98
6.2	PHASE 1 - ANALYSE	100
6.2.1	ASCI pour le pilotage hybride	100
6.2.2	Sous-Système Physique (SSP)	101
6.2.3	Sous-Système Logique (SSL)	103
6.2.4	Sous-Système Décisionnel (SSD)	103
6.3	PHASE 1 - SPÉCIFICATION	105
6.3.1	Diagramme d'activité générale	105
6.3.2	Diagramme d'activité représentant la préparation d'une livraison	106
6.3.3	Diagramme d'activité représentant la livraison et le rangement d'une livraison de doses de médicament	107
6.4	PHASE 1 - ENTITÉS DÉCISIONNELLES ET TRANSFORMATION 2 DE LA DÉMARCHE MDA-RHMS	109
6.4.1	Le SIH	109
6.4.2	Les diagrammes des entités décisionnelles	109
6.4.2.1	Isolation des holons	109
6.4.2.2	Représentation d'une dispensation globale et globalisée	110
6.4.2.3	Représentation d'une dispensation globale et globalisée	111
6.5	PHASE 1 - BILAN	112
6.5.1	La démarche de réingénierie d'un circuit du médicament	112
6.5.2	Création d'une bibliothèque de composants	113
6.6	PHASE 2 - CONCEPTION ET IMPLÉMENTATION : CAS D'ÉTUDE D'UNE INTERFACE DE LOGICIEL DE SIMULATION	113
6.6.1	Conception	113
6.6.2	Implémentation	115
6.7	PHASE 2 - CONCEPTION ET IMPLÉMENTATION : CAS D'ÉTUDE DU SUIVI DU CIRCUIT DU MÉDICAMENT PAR IDENTIFICATION RADIOFRÉQUENCE (RFID)	116
6.7.1	Cahier des charges	116

6.7.2	Conception	118
6.7.3	Implémentation	121
6.7.3.1	Choix des logiciels de programmation	121
6.7.3.2	Préparation	122
6.7.3.3	Administration	123
6.8	PHASE 2 - CONCEPTION ET IMPLÉMENTATION : CAS D'ÉTUDE DE L'UTILISATION DU LOGICIEL DE SIMULATION AUTO MOD	124
6.8.1	État de l'art sur la simulation du circuit du médicament	124
6.8.2	Bilan et perspectives sur la simulation du CDM	125
6.8.3	Le logiciel de simulation Automod	125
6.8.3.1	Présentation de Automod	125
6.8.3.2	Resources	126
6.8.3.3	Load	126
6.8.3.4	Queue	126
6.8.3.5	Système de déplacement	126
6.8.3.6	Order list	127
6.8.4	Simulation d'une pharmacie générique	127
6.8.4.1	Pharmacie générique pour une livraison globale	127
6.8.4.2	Pharmacie générique pour une livraison nominative	128
6.8.4.3	Représentation des stocks de médicaments	130
6.8.4.4	Interface d'entrée-sortie	130
6.9	PHASE 2 - BILAN	131
6.10	GÉNÉRICITÉ DE L'APPROCHE PROPOSÉE	131
6.11	CONCLUSION	132
7	INSTANCIATION SUR UN SYSTÈME : LE CHU DE CLERMONT-FERRAND	135
7.1	INTRODUCTION	137
7.2	CARACTÉRISTIQUES D'UN SYSTÈME	139
7.2.1	Définition des caractéristiques du système	139
7.2.2	Le langage utilisé	139
7.2.3	À partir des diagrammes	141
7.2.4	À partir des matrices de déroulement	141
7.2.5	Étude de scénario avec la pharmacie nominative générique	143
7.3	MODÈLE DE CONNAISSANCE DU SYSTÈME	144
7.3.1	Le SIH au CHU de Clermont-Ferrand	144
7.3.2	Modèles pour la réingénierie à partir du MGC	146
7.3.2.1	Construction du Modèle de Connaissance	146
7.3.2.2	Fonctionnement actuel du CDM au CHU de Clermont-Ferrand	146
7.3.2.3	Fonctionnement souhaité du CDM au CHU de Clermont-Ferrand	146
7.4	MODÈLE D'ACTION	148
7.4.1	Élaboration du modèle d'action	148
7.4.2	Un nouvel hôpital	149
7.4.2.1	L'hôtel Dieu	149
7.4.2.2	Le CHU Estaing	149
7.4.3	Modélisation du CHU Estaing	151
7.4.3.1	La distribution	151
7.4.3.2	Rangement des UF	152
7.4.3.3	La pharmacie du CHUE	152

7.4.4	Interface d'entrée	153
7.4.4.1	Hypothèses de fonctionnement	153
7.4.4.2	Listes des doses à préparer et évaluation du volume des médicaments transportés	154
7.4.4.3	Attribution des UF par tournée	155
7.4.5	Ajout de la pharmacie générique pour une livraison nominative	155
7.4.6	Bilan et exploitation du modèle d'action	157
7.4.6.1	Correspondance avec le MC et perspectives du modèle d'action	157
7.4.6.2	Cadre expérimental	158
7.5	MODÈLE DE RÉSULTATS	158
7.5.1	Mise en œuvre du modèle d'action	158
7.5.2	Alimentation du modèle de résultats	160
7.6	CONCLUSION	162
	CONCLUSION GÉNÉRALE ET PERSPECTIVES	163
	BIBLIOGRAPHIE	169
	A ANNEXES	181
A.1	LE CIRCUIT PNEUMATIQUE DU CHU ESTAING ET SA REPRÉSENTATION SOUS AUTOMOD	182
A.1.1	Le circuit pneumatique utilisé au CHU Estaing	182
A.1.2	Le synoptique des circuits pneumatiques et leur représentation dans Automod	182
A.2	COMPLÉMENT D'INFORMATION SUR LE SUIVI DU CIRCUIT DU MÉDICAMENT PAR IDENTIFICATION RADIOFRÉQUENCE (RFID)	184
A.2.1	Programmation des tags	184
A.2.2	Exemple de code pour l'écriture d'un tag en C#	185
A.3	EXPLICATION DU CODE AUTOMOD POUR UNE PHARMACIE GÉNÉRIQUE AVEC UNE DISPENSATION NOMINATIVE	186
A.3.1	Création des boîtes correspondant aux patients	186
A.3.2	Choix du parcours de la boîte du patient	186
A.3.3	Arrivée dans le stock du service	188
A.4	DÉTAIL DE L'IMPLANTATION D'UN MODÈLE DANS AUTOMOD	189
A.4.1	Comment implanter un sous modèle	189
A.4.2	Modifications apportées au modèle de l'hôpital Estaing	189

LISTE DES FIGURES

1.1	Exemples de automates « toutes formes » de préparation nominative	10
1.2	Automates « formes orales sèches » PACMED avec des pilules ensachées	10
1.3	Description des entités du circuit du médicament pour une dispensation nominative [Huet 10a]	11
1.4	Différentes préparations de robots (à droite SWISSLOG, les autres SINTECO)	13
1.5	Exemple de système d'assemblage : une carte électronique	14

1.6	Exemple d'un système d'assemblage en série [Fontanili 99]	15
1.7	Exemple d'un système d'assemblage avec dérivation [Pétin 07]	16
1.8	Exemple d'un système d'assemblage sans dérivation [Heverhagen 01]	16
2.1	Exemples de structures de pilotage [Cardin 07]	22
2.2	Balance entre différents types de pilotage suivant la flexibilité d'après [Bongaerts 00]	24
2.3	Architecture générale d'un holon [Mcfarlane 00]	24
2.4	Bloc de construction basique d'un HMS et leurs relations [Brussel 98, Valckenaers 98]	25
2.5	Un HMS et ses holons de base avec un diagramme de classe [Valckenaers 98]	26
2.6	Les différents niveaux de décision d'une entreprise	28
2.7	Architecture de décision centrée sur le produit inspirée par [Morel 07, Pannequin 07, Auzelle 09]	31
2.8	Interopérabilité dirigée par le produit [Auzelle 08]	32
2.9	Concept d'interopérabilité appliqué au SIP et au CDM	33
2.10	Classification de l'interopérabilité orientée données [Tolk 03]	35
3.1	Le cadre de modélisation de CIMOSA	42
3.2	Les différents niveaux des modèles de la démarche MDA [Frankel 03]	44
3.3	La méthodologie ASCI	46
3.4	Processus d'analyse bidirectionnel [Chiron 09]	47
3.5	Les phases de MaSE [Ali 09]	48
3.6	OPEN Process Framework [Garcia-Ojeda 08]	49
3.7	La méthodologie Gaia [Ali 09]	50
3.8	Analyse descendante systémique avec IDEFO/SADT	52
3.9	Vue Générale d'un Bloc Fonction (FB) [Chiron 09]	55
3.10	Exemple d'un processus d'assemblage [Fontanili 99]	59
3.11	Bibliothèque des différentes entités transitiques de [Paris 93]	60
3.12	Exemple d'application avec une cellule composée de deux postes simples [Paris 93]	60
4.1	Modèle conceptuel pour les holons ADACOR [Leitão 06]	64
4.2	Interaction entre les Product Holon et les Task Holons avec un diagramme d'interaction AUML [Leitão 04]	65
4.3	Les principales « Holonic class » d'après [Simão 05]	66
4.4	Spécialisation des holon-ressources d'après [Simão 05]	66
4.5	L'approche par quatre niveaux ontologiques	68
4.6	Méta-modèle holonique pour la représentation du produit d'après [Baïna 06b]	68
4.7	Les différents niveaux ontologiques utilisés par [El Haouzi 08a]	69
4.8	Enrichissement du lien entre les différents sous-systèmes dans la phase d'Analyse d'ASCI	70
5.1	Le « pattern composite » inspiré de [Gamma 95]	74
5.2	Transformation de modèle	75
5.3	Vue d'ensemble de la relation entre UML et SysML inspirée par [OMG 10a]	76
5.4	Le Méta-Méta-Modèle décrivant les relations de base dans un système holonique de production	78
5.5	Transformation entre le niveau M3 et M2	80

5.6	Méta-Modèle MM ₁ , montrant les relations dans un système de production contrôlé par les ressources	81
5.7	Méta-Modèle MM ₂ , montrant les relations dans un système de production contrôlé par le produit	82
5.8	Comparaison entre l'approche montante de MBCSA et l'étude d'un domaine avec ASCI	85
5.9	Comparaison entre l'approche descendante de MBCSA et l'étude d'un système avec ASCI	86
5.10	Méthodologie ASCI-RHMS [Huet 10c]	88
5.11	Passage du Méta-Modèle au Modèle Générique de Connaissance d'un domaine	90
5.12	Passage du Modèle Générique de Connaissance d'un domaine au Modèle de Connaissance d'un système du domaine	91
5.13	MRHMS et ses interactions méthodologiques internes entre MDA-RHMS et ASCI-RHMS	93
6.1	Positionnement du chapitre 6 dans le mémoire	98
6.2	Construction d'une bibliothèque de composants réutilisables	99
6.3	Démarche de modélisation pour la conception d'un MGC du CDM	100
6.4	Sous-Système Physique de ASCI-CDM	102
6.5	Sous-Système Logique de ASCI-CDM	103
6.6	Sous-Système Décisionnel de ASCI-CDM	104
6.7	Diagramme d'activité représentant le comportement général du circuit du médicament	105
6.8	Diagramme d'activité représentant une préparation de doses	106
6.9	Diagramme d'activité représentant une livraison et un rangement de doses	108
6.10	Place du Système d'Information Pharmaceutique dans le Système d'Information Hospitalier	109
6.11	ED pour un fonctionnement globalisé	110
6.12	ED pour un fonctionnement nominatif	111
6.13	Diagramme d'activité technique pour la conception d'un chariot	114
6.14	Interface utilisateur Excel de l'interface entrée/sortie de logiciels de simulation	115
6.15	Identification des sachets dans la pharmacie	118
6.16	Choix du moyen de transport dans la pharmacie	119
6.17	Préparation de la distribution des sachets dans un service	119
6.18	Administration des médicaments au patient	120
6.19	Diagramme relationnel de la base de données géré par le dispositif	121
6.20	Exemple d'affichage après l'écriture du tag à la pharmacie	122
6.21	Exemple d'affichage après la lecture pour la validation au lit du malade	123
6.22	Exemple d'une pharmacie en fonctionnement global du circuit du médicament	127
7.1	Positionnement du chapitre 7 dans le mémoire	137
7.2	Modélisation possible du circuit du médicament	141
7.3	Scénario pour l'étude de la pharmacie générique	143
7.4	Capture d'écran de la pharmacie générique avec le scénario étudié	144
7.5	Capture d'écran du logiciel de prescription informatisée	145
7.6	Capture d'écran du logiciel de gestion des prescriptions au niveau de la pharmacie	145
7.7	Modèle de la dispensation globalisée au CHU de Clermont-Ferrand	147
7.8	Modèle d'une dispensation nominative au CHU de Clermont-Ferrand	148

7.9	Les différents niveaux du CHU Estaing [Rodier 10]	150
7.10	Livraison des UF du CHU Estaing	152
7.11	Zone de préparation de la pharmacie globale	153
7.12	Modèle complet avec une pharmacie avec une livraison nominative (1), une pharmacie avec une livraison globale (2), un circuit pneumatique (3) et un circuit chariot (4)	156
7.13	Modèle de résultats pour un scénario avec une dispensation nominative	161
7.14	Modèle de résultats pour un scénario avec une dispensation globale	161
A.1	Cartouche et stations d’envoi et de réception du circuit pneumatique	182
A.2	Synoptique représentant un des trois circuits pneumatiques au CHU Estaing de Clermont-Ferrand	183
A.3	Modélisation dans Automod du circuit pneumatique	183
A.4	structure de la mémoire des tags utilisés	184
A.5	Extrait du code pour la vérification de la concordance au lit du malade	185
A.6	Code Automod pour la création d’une boîte qui contiendra les DPP	186
A.7	Code Automod pour la lecture des ordonnances	187
A.8	Code Automod pour le choix du parcours lors de l’assemblage des DPP	187
A.9	Code Automod pour le rangement des DPP dans le stock des pôles et des services.	188
A.10	Arbre des modèles sous Automod	189

LISTE DES TABLEAUX

1.1	Description de l’analogie avec les ressources utilisées [Huet 09b, Huet 09c]	13
2.1	Modèle de référence de LISI [AWG 97]	34
3.1	Le cadre de Zachman	41
3.2	Les quatre niveaux d’architecture de TROPOS [Bresciani 04]	50
3.3	Signification des symboles de la norme JIS Z 8206	58
3.4	Les quatre symboles de base de [Bourrières 90]	59
5.1	Démarche MDA préconisée par notre méthodologie	75
5.2	Comparaison des facettes de composants vues par [Chiron 09] avec un holon vu par [Mcfarlane 00]	83
5.3	Comparaison de l’approche montante de MBCSA et d’ASCI	86
5.4	Comparaison de l’approche descendante de MBCSA et d’ASCI	86
6.1	Actions autorisées pour chaque acteur	107
6.2	Différents types de chariot	114
6.3	Récapitulatif de la représentation des services, des pôles et des patients sous Automod, en DJIN et en dispensation globale	129

6.4	Récapitulatif de la représentation des services, des pôles et des patients sous Automod, en DJIN et en dispensation globale/globalisée	130
7.1	Signification des symboles de la norme JIS Z 8206 et leur correspondance sur le circuit du médicament	140
7.2	Matrice de déroulement du chemin principal sans aucune opération d'assemblage	142
7.3	Matrice de déroulement pour atteindre une zone de stockage	142
7.4	Matrice de déroulement pour une opération d'assemblage	142
7.5	Récapitulatif des hypothèses	154
7.6	Choix des paramètres d'entrée	159
7.7	Ressources représentées dans le modèle de résultats du scénario utilisé	160
7.8	Approche développée dans cette thèse	165

LISTE DES ACRONYMES

ADACOR	ADaptive holonic COntrol aRchitecture for distributed manufacturing systems
API	Automate Programmable Industriel
ARIS	ARchitecture de Systèmes d'Information Intégrés
ASCI	Analyse, Spécification, Conception, Implémentation
ASCI-CDM	ASCI pour la Réingénierie du CDM
ASCI-PH	ASCI pour le Pilotage Hybride
ASCI-RHMS	ASCI pour la Réingénierie des HMS
AUML	Agent Unified Modeling Langage
BDD	Block Definition Diagram - diagramme de définition de bloc
BPM	Business Process Modeling - gestion des processus métier
CDM	Circuit Du Médicament
CHU	Centre Hospitalier Universitaire
CIMOSA	CIM Open System Architecture
DJIN	Dispensation Journalière Individuelle Nominative
DPP	Doses Préparées pour un Patient
DPS	Doses Préparées pour un Service
DMS	Dispositifs Médicaux Stériles
ED	Entités Décisionnelles
ERP	Entreprise Ressource Planning - Progiciel de gestion intégré
FB	Function Block
FMS	Flexible Manufacturing System - Système manufacturier flexible
GIM	GRAI Integrated Methodology
GRAI	Graphes à Résultats et Activités Inter-reliés
HCBA	Holonic Component-Based Approach
HMS	Holonic Manufacturing System
HO	Holon-Ordre
HP	Holon-Produit
HR	Holon-Ressource
HS	Holon-Staff
IBD	Internal Block Diagram - diagramme interne de bloc
IDE	Infirmier Diplômé d'État

IDEF	Integrated computer aided manufacturing DEFinition language
IDM	Ingénierie Dirigée par les Modèles
IMS	Intelligent Manufacturing Systems
LCIM	Levels of Conceptual Interoperability Model
LISI	Levels Information System Interoperability
LOTOS	Language Of Temporal Ordering Specification
MARTE	Modeling and Analysis of Real-Time and Embedded systems
MaSE	Multiagent Systems Engineering
MBCSA	Méthodologie Bidirectionnelle de Conception des Systèmes Automatisés
MC	Modèle de Connaissance
MDA	Model Driven Architecture - Architecture dirigée par les modèles
MDA-RHMS	MDA pour la Réingénierie des HMS
MERISE	Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise
MES	Manufacturing Execution System
MGC	Modèle Générique de Connaissance
MRHMS	Méthodologie de Réingénierie des HMS
O-MaSE	Organization-based Multiagent System Engineering
OMG	Object Management Group
PERA	Purdue Enterprise Reference Architecture
PIM	Platform Independent Model
PROSA	Product-Resource-Order-Staff Architecture
PSM	Platform Specific Model
PROSIS	Product, Resource, Order and Simulation Isoarchic Structure
PUI	Pharmacie à Usage Interne
RdP	Réseaux de Petri
RFID	Radio Frequency IDentification
SCP	Système Contrôlé par le Produit
SdS	Système de Systèmes
SI	Système d'Information
SIE	Système d'Information d'Entreprise
SIH	Système d'Information Hospitalier
SIP	Système d'Information Pharmaceutique
SMA	Système Multi-Agent
SSD	Sous-Système Décisionnel
SSL	Sous-Système Logique
SSP	Sous-Système Physique
SysML	Systems Modeling Language - Langage de Modélisation des systèmes
UEML	Unified Enterprise Modeling Langage - langage unifié de modélisation d'entreprise
UF	Unité Fonctionnelle
UML	Unified Modeling Language - Langage de modélisation unifié

INTRODUCTION GÉNÉRALE

LES hôpitaux subissent actuellement des changements majeurs dans leur structure et dans leurs opérations, dues aux changements démographiques, à l'augmentation de la compétition parmi les hôpitaux et les prestataires, et à une augmentation des coûts en équipement et en technologie de l'information. Ces changements déstabilisent la façon dont les hôpitaux ont l'habitude de travailler et leurs économies. Les dirigeants hospitaliers ont maintenant besoin d'optimiser ces systèmes de « production » pour prendre en compte les nouvelles réalités des demandes des clients et de l'état. Les dirigeants et les praticiens hospitaliers ainsi que les chercheurs s'intéressent à différents problèmes relevant de ce domaine.

Le travail présenté dans ce mémoire de thèse a été réalisé dans le cadre d'une convention de recherche entre le LIMOS et le *Centre Hospitalier Universitaire (CHU)* de Clermont-Ferrand qui a financé plusieurs thèses [Chauvet 09, Rodier 10, André 10] sur le projet de modélisation, de simulation et d'optimisation des flux du CHUE (CHU Estaing qui est opérationnel depuis avril 2010). Ce projet a été lancé en partant du constat que le milieu hospitalier se réorganise et cherche à utiliser des méthodes nouvelles pour permettre une réduction des coûts et une amélioration des soins.

Cette thèse s'intéresse plus particulièrement au *Circuit Du Médicament (CDM)*. Ce circuit est composé d'un ensemble de trois étapes : la prescription du médecin, la dispensation de doses préparées à la pharmacie et l'administration de doses par l'infirmier au malade. Le CDM est un élément primordial de l'organisation des établissements de santé. Sa sécurisation et sa modernisation permettent de réduire le nombre d'erreurs médicamenteuses (iatrogénie médicamenteuse) et d'optimiser les coûts de fonctionnement.

La recommandation officielle du document « Information du circuit du médicament dans les établissements de santé » [DHOS 01] indique la nécessité de réduire la quantité de temps de travail passé par les infirmières dans l'administration des médicaments et de réduire leur consommation. L'automatisation de la dispensation et les systèmes robotisés dans les *Pharmacies à Usage Interne PUI* prolifèrent depuis les années 1990, en partie à cause de la nécessité de limiter les coûts et d'étendre les fonctions des pharmaciens, pour les faire évoluer de la simple fonction de distribution des médicaments vers celle d'une analyse pharmaceutique. L'automatisation du circuit du médicament dans un hôpital inclut des systèmes centralisés de remplissage des chariots, des systèmes décentralisés localisés dans les unités de soin et un système d'information [Arriudarre 98]. L'utilisation des robots et autres automates de dispensation dans une PUI devrait améliorer les soins cliniques, réduire les coûts et améliorer la précision. Les coûts seront réduits grâce à la baisse du nombre d'erreurs médicamenteuses (diminution du nombre de personnes blessées et décédées) et une meilleure utilisation du stock de médicaments. En effet, ces optimisations du CDM amélioreront le respect des cinq règles concernant l'administration des médicaments qui sont : le bon médicament, la bonne dose, la bonne forme, au bon moment, pour le bon patient.

Parallèlement à ces problématiques propres au milieu hospitalier, des recherches récentes ont orienté les entreprises vers des structures distribuées où les décisions sont prises locale-

ment par des entités dotées de capacité de perception et d'action, ce qui permet d'augmenter la réactivité du système. Cependant, celles-ci posent le problème de la cohérence globale des décisions que cette autonomie peut mettre en péril. De nouvelles structures sont alors apparues comme un compromis entre une structure purement distribuée et une structure centralisée pour assurer autonomie et réactivité avec cohérence globale [El Haouzi 08b]. C'est dans ce cadre que le concept « Holonique » présenté par [Koestler 67] puis, plus tard, le paradigme du *Système Contrôlé par le Produit* (SCP) sont apparus [Gouyon 08]. Ce problème de cohérence globale entre les structures de décision fait apparaître différents niveaux de coordination à différentes échelles entre les différents SI composant une entreprise, faisant émerger des problèmes d'« interopérabilité » et de nouvelles visions de l'entreprise comme celle de *Système de Systèmes* (SdS) [Auzelle 08].

Nous allons montrer dans ce manuscrit que ces problématiques sont également présentes sur le CDM d'un hôpital. Nous démontrerons que le concept de SCP est adapté à la représentation d'un CDM où les doses préparées pour chaque patient sont équipées avec des technologies d'identification. Nous tenterons alors d'utiliser les méthodologies développées dans le domaine des systèmes de production pour réorganiser le CDM, nous expliquerons les manques des méthodologies actuelles que nous avons remarqués.

Nous proposerons alors une nouvelle méthodologie développée dans le but d'avoir un contrôle par le produit des systèmes manufacturiers afin de permettre la réingénierie d'un CDM. Cette méthodologie sera applicable à tout *Holonic Manufacturing System* (HMS). Nous l'instancierons ensuite sur le domaine du CDM, puis sur un système de ce domaine avec un exemple d'hôpital : le CHU de Clermont-Ferrand.

Pour permettre cette réingénierie, nous proposerons un outil d'aide à la décision. Cet outil a permis d'organiser au mieux le CHU Estaing de Clermont-Ferrand qui a ouvert en avril 2010 et permettra aux pharmaciens de réorganiser la pharmacie centrale qui sera refaite à neuf dans quelques années. Pour concevoir un tel outil, nous avons besoin d'une méthodologie de modélisation qui analysera le système tel qu'il est et tel que nous souhaiterions qu'il soit.

Le *premier chapitre* de ce mémoire situe le contexte et les problématiques. Il présente le CDM et les enjeux de sa réingénierie ; nous détaillerons en quoi un CDM et un système de production sont similaires, ainsi que l'intérêt d'une telle analogie.

Le *deuxième chapitre* présente en détail le concept de contrôle par le produit de systèmes manufacturiers. Nous exposerons les travaux récents effectués autour de ce concept comme ceux autour de la notion d'interopérabilité, de *Système de Systèmes* (SdS) ou de *Holonic Manufacturing System* (HMS).

Le *troisième chapitre* présente un état de l'art sur les outils de modélisation relatifs aux systèmes de production. Afin d'effectuer nos choix, nous exposerons les cadres d'architecture, les méthodologies, les méthodes et les langages qui nous semblent les plus intéressants pour modéliser et réorganiser un système de production de biens ou de services.

Le *quatrième chapitre* présente les outils utilisés plus précisément pour la modélisation et la conception des systèmes holoniques. Il reprend les outils présentés au chapitre 3 et expose ceux qui ont été utilisés pour l'étude des HMS.

Le *cinquième chapitre* expose en détail notre proposition méthodologique en justifiant les différents choix que nous avons effectués concernant les outils décrits aux chapitres 3 et 4.

Cette méthodologie sera utilisable pour la réingénierie de n'importe quel **HMS**.

Le *sixième chapitre* présente une application de notre méthodologie sur le domaine du **CDM**. Nous détaillerons les étapes suivies pour la construction du *Modèle Générique de Connaissance (MGC)* qui prend en compte tous les fonctionnements possibles du **CDM** en France. Nous avons construit une bibliothèque de composants réutilisables, par exemple un dispositif de vérification au lit du malade à partir de puces **RFID** ajoutées aux *Doses Préparées pour un Patient (DPP)*(le produit).

Dans le *septième chapitre*, nous proposerons de reprendre le **MGC** relatif au domaine du **CDM** construit dans le chapitre 6 et de l'instancier sur un système avec la mise en place d'un scénario au **CHU** de Clermont-Ferrand. A partir du **MC** qui découlera de cette instanciation, nous générerons un modèle d'action et un modèle de résultats qui pourront être utilisés comme outil d'aide à la décision par les pharmaciens et les managers de l'hôpital. Cet outil d'aide à la décision a été élaboré par l'intermédiaire d'un logiciel de simulation.

La conclusion dresse un bilan de la contribution de cette thèse et présente plusieurs perspectives à ces travaux de recherche.

1

CONTEXTE ET PROBLÉMATIQUES

RÉSUMÉ DU CHAPITRE 1

Dans ce chapitre, nous définissons les différentes étapes du circuit du médicament et les enjeux de sa réorganisation pour la santé publique ainsi que les gains financiers qui peuvent en découler. Nous détaillons le rôle de chaque acteur sur ce circuit et la façon dont chacun de ces acteurs peut améliorer son fonctionnement à l'aide de nouvelles technologies. Parmi ces technologies, on peut citer des robots de dispensation, des armoires sécurisées, l'utilisation des technologies d'identification et de nouveaux logiciels venant enrichir le système d'information pharmaceutique dans toutes les étapes du circuit du médicament. Nous détaillons ensuite la complexité du système d'information d'un hôpital qui est confronté aux mêmes problématiques que celui d'une entreprise. On se retrouve face à la problématique où l'on doit synchroniser un flux d'information avec le flux physique des médicaments. Enfin, nous faisons une analogie entre le circuit du médicament et un système d'assemblage : nous pouvons alors utiliser les travaux récents associés à ce domaine.

SOMMAIRE

1.1	INTRODUCTION	6
1.2	LE CIRCUIT DU MÉDICAMENT DANS UN HÔPITAL	6
1.2.1	Fonctionnement du circuit du médicament	7
1.2.2	Objectif : la dispensation nominative	8
1.3	SYNCHRONISATION DES FLUX PHYSIQUES ET INFORMATIONNELS	12
1.3.1	Synchronisation des flux du CDM	12
1.3.2	Interopérabilité avec le Système d'Information Hospitalier	12
1.3.3	Interaction avec le flux physique	12
1.4	ANALOGIE ENTRE LE CDM ET UN FMS	14
1.5	CONCLUSION	17

1.1 INTRODUCTION

Ce chapitre décrit le contexte de cette étude et les enjeux de la réingénierie du *Circuit Du Médicament* (CDM) dans un établissement de santé. Nous commencerons par détailler le fonctionnement du CDM puis nous expliquerons l'analogie que nous proposons avec un système flexible de production manufacturier. Cette analogie a pour but de pouvoir appliquer les travaux récents sur les systèmes manufacturiers flexibles sur le CDM. L'objectif est l'optimisation du CDM pour améliorer le respect des cinq règles concernant l'administration des médicaments dans un établissement hospitalier qui sont : le bon médicament, la bonne dose, la bonne forme, au bon moment, pour le bon patient. La première section sera consacrée au détail du fonctionnement du CDM dans un hôpital, la seconde aux enjeux de la synchronisation des flux physiques et informationnels et enfin, sa correspondance avec le milieu industriel sera expliquée.

1.2 LE CIRCUIT DU MÉDICAMENT DANS UN HÔPITAL

Le circuit du médicament est sous la responsabilité des pharmaciens, le stock de médicaments est géré par une *Pharmacie à Usage Interne* (PUI), la pharmacie de l'hôpital. Au CHU de Clermont-Ferrand, le flux de médicaments représentait 70 M€ de budget en 2008 avec 2200 médicaments actifs. C'est un des budgets les plus importants du CHU.

En premier lieu, nous proposons deux définitions du CDM qui se complètent. La définition 1.1 émise par la mission ministérielle de la MEAH [Cauterman 07] est plus centrée sur le personnel et la définition 1.2, qui vient du *dictionnaire de l'erreur médicale* [Schmitt 06], montre bien la complexité du *Système d'Information* (SI) qui va en découler.

Définition 1.1 *En pratique, le circuit du médicament dans les hôpitaux et les cliniques est un macro-processus de soins complexe, transversal, jalonné d'étapes mobilisant des professionnels différents : la prescription est un acte médical, la dispensation, un acte pharmaceutique et l'administration, un acte infirmier ou médical [Cauterman 07].*

Définition 1.2 *Le circuit du médicament est défini comme étant le processus de la prise en charge thérapeutique médicamenteuse d'un patient, hospitalisé dans un établissement de santé, résident d'un établissement social ou en soins ambulatoires. Ce processus interdisciplinaire est constitué par les étapes de prescription, de dispensation, d'administration et de suivi thérapeutique, et comprend le traitement de l'information [Schmitt 06].*

On remarque donc que 3 acteurs sont concernés par le CDM :

- le médecin lors de la prescription,
- le couple pharmacien/préparateur pour la dispensation,
- l'infirmier lors de l'administration au patient.

Le circuit hospitalier du médicament est considéré comme l'un des processus les plus transversaux et les plus critiques dans un établissement de santé. Chaque étape du circuit (la prescription, la dispensation et l'administration), est source d'erreurs potentielles pouvant mettre en jeu la vie du patient [Ghorfi 10].

La sécurisation et la modernisation du CDM ont pour objectif de permettre de baisser sensiblement le nombre de personnes victimes d'erreurs médicamenteuses (iatrogénie médicamenteuse) et de réduire les coûts (augmentation du rendement, libération de postes préparateurs et gain de temps infirmier. . .). Il est difficile d'évaluer l'impact de ces erreurs mais les chiffres les plus généralement cités en France concernant l'iatrogénie médicamenteuse font état d'au moins 13 000 morts par an et 140 000 hospitalisations¹ et d'environ 100 000 morts par an aux USA [Starfield 00]. L'erreur médicamenteuse se produit généralement à l'un des quatre niveaux suivants : prescription, transcription, délivrance ou administration [Bedouch 09].

A côté de l'évolution de ces 3 métiers : protocolisations pluridisciplinaires, mise en place de l'évaluation des pratiques professionnelles, développement de la pharmacie clinique et de la présence pharmaceutique dans les services, la sécurisation et la modernisation du CDM passent également par de nouvelles organisations du circuit et l'utilisation de nouvelles technologies.

Les nouvelles technologies quant à elles permettent :

- d'aider le médecin, le pharmacien et l'infirmier dans leurs décisions notamment via l'informatisation du circuit,
- d'automatiser la dispensation à la pharmacie ou dans les services, ce qui réduit encore les taux d'erreurs et rend possible sa mise en place avec un effectif de préparateurs constant,
- d'automatiser le transport des médicaments vers les unités de soins via les technologies logistiques (pneumatique, chariots autoguidés, caissons sur monorails),
- de sécuriser l'administration au lit du patient via les techniques de traçabilité (Code-barres, RFID),
- d'aider les décideurs dans les choix organisationnels, technologiques et financiers via les techniques de modélisation et de simulation.

1.2.1 Fonctionnement du circuit du médicament

Afin de comprendre le fonctionnement du CDM en général sans se focaliser sur le CHU de Clermont-Ferrand, nous nous sommes basés sur la littérature pharmaceutique [Arriudarre 98, Cairns 01, Madic 01, Labrosse 06] et la littérature sur les systèmes de production qui traite ce sujet [Majchrzak 06, Aloui 07, Di Martinelly 08]. Nous avons également visité des hôpitaux français. Ce CDM est sous la responsabilité des pharmaciens et a pour objectif l'envoi de médicaments à destination des patients et/ou des services. Devant la complexité du CDM Nous avons limité l'étude dans ce manuscrit aux trois types principaux de produits pharmaceutiques : les petits médicaments (gélules, sachets, . . .), les solutés (le liquide servant de vecteur) et les *Dispositifs Médicaux Stériles* (DMS). En effet, L'expression « circuit du médicament » est le terme consacré, mais il est possible d'étendre la notion à l'ensemble des produits pharmaceutiques : gaz médicaux, dispositifs médicaux. . .

Définition 1.3 *La dispensation regroupe les activités pharmaceutiques induites par la prescription médicale depuis l'analyse de l'ordonnance jusqu'à l'acheminement du médicament dans l'unité de soins.*

1. Audition au sénat de MM. Claude Huriet, président, et Dominique Martin, directeur de l'Office national d'indemnisation des accidents médicaux, des affections iatrogènes et des infections nosocomiales (Oniam) : <http://www.senat.fr/commission/soc/medicamento51212.html> (accessible au 23 juillet 2010)

La dispensation (définition 1.3) peut être de trois types :

- **nominative**, la prescription du médecin est communiquée au pharmacien qui l’analyse, puis les doses de médicament sont préparées séparément pour chaque patient par le personnel de la pharmacie et livrées dans l’unité de soins ; c’est l’organisation la plus sécurisée,
- **globalisée**, la prescription du médecin est communiquée au pharmacien qui l’analyse, puis les doses de médicament sont additionnées par médicament par le personnel de la pharmacie et livrées dans l’unité de soins,
- **globale**, l’infirmier de l’unité de soins commande les médicaments en croisant les prescriptions du médecin et ce qu’il reste de sa dotation, les doses demandées sont préparées par le personnel de la pharmacie et livrées dans l’unité de soins. Ce type de dispensation est actuellement le principal mode utilisé au **CHU** de Clermont-Ferrand. Il peut être optimisé par la mise en place d’un système de type plein/vide, analogue au système KANBAN, facilitant la commande et la gestion du stock par l’infirmier [Laborie 06].

A noter que cette division en trois types du terme « dispensation » fait débat parmi les pharmaciens, certains considèrent que le terme « dispensation » se rapporte seulement à la dispensation nominative et globalisée et dans ce cas, ce que nous avons appelé « dispensation globale » est appelé « distribution ». Il y a alors « dispensation » seulement quand il y a une validation pharmaceutique. Ainsi, par exemple pour la norme d’interopérabilité avec le **CDM PN13** [Roux 10b], les trois types sont : dispensation nominative, dispensation reglobalisée et distribution.

Au **CHU** de Clermont-Ferrand, la majorité des services fonctionnent avec une dispensation globale mais plusieurs services testent d’autres types de fonctionnement en partenariat avec la pharmacie :

- Un service de cardiologie teste la dispensation globalisée avec prescription informatisée, analyse pharmaceutique de chaque prescription et validation informatique de chaque administration au patient par l’infirmier. C’est une première étape avant le passage à la dispensation nominative.
- Un service de chirurgie digestive teste une dispensation globale en système plein/vide avec déclenchement de la commande par puces **RFID**. Ce type de système restera pour les produits qui ne pourront être gérés en nominatif.

1.2.2 Objectif : la dispensation nominative

La dispensation nominative est aujourd’hui l’organisation qui a montré les meilleurs résultats en termes de sécurité et est recommandée par le ministère [DHOS 01, loi 99]. En effet, la dispensation nominative, où, après analyse de la prescription du médecin par le pharmacien, les médicaments sont préparés et délivrés par du personnel pharmaceutique pour chaque patient, est l’organisation du circuit qui réduit le plus le risque d’erreurs. Elle permet un gain de temps infirmier et un gain de stocks dans les services de soins. Lorsque sa réalisation est manuelle, elle nécessite un effectif de préparateurs important. La dispensation nominative peut avoir plusieurs fréquences d’utilisation. La plus répandue est l’envoi journalier appelé la *Dispensation Journalière Individuelle Nominative* (**DJIN**) mais peut être aussi hebdomadaire (**DHIN**). Les difficultés de généralisation de la **DJIN** résident dans [Labrosse 06] :

- Le délai entre la prescription et l’administration. En effet, la préparation à la **PUI** des traitements nominatifs nécessite un délai entre la prescription et l’administration,
- Les moyens humains nécessaires. Selon un rapport publié en 2001 par le ministère délé-

gué à la santé, un ordre de grandeur allant de 1/4 à 1 Equivalent Temps Plein préparateur est nécessaire à la préparation en **DJIN** manuelle par unité de soins.

Pour pallier ces difficultés, différentes solutions d'automatisation sont proposées. Les nouvelles technologies nécessitent une prescription informatisée pour une utilisation optimale. L'automatisation permet la transmission des données de prescription validées pharmaceutiquement à un automate dont les déplacements vont faciliter ou réaliser la cueillette (picking) des médicaments et leur répartition individuelle nominative [Labrosse 06]. Des études indépendantes américaines ont déjà permis de montrer les gains financiers et humains qu'apportent les nouvelles technologies dans ce secteur [Shack 06] : par exemple, dans le centre médical de Bowling Green au État-Unis, ils estiment un retour sur investissement au bout de 5 ans après une automatisation.

On peut différencier plusieurs types d'automates :

- Les automates de cueillette appelés aussi armoires automatisées ou stockeurs rotatifs. Ils sont une aide à la dispensation nominative effectuée manuellement par les préparateurs dans la **PUI**. Ces automates sont retrouvés également dans les officines. Ils assurent un stockage important dans un minimum d'espace et permettent une meilleure organisation du stockage.
- Les automates de dispensation de formes orales sèches (gélules) centralisés au niveau de la **PUI** permettent le conditionnement des médicaments dans des sachets individuels en fonction des prescriptions. Ces automates nécessitent l'approvisionnement en médicaments sans blisters (emballage des médicaments) et imposent une étape de déblistériation pour obtenir des formes vrac. Les médicaments n'étant plus dans leur emballage d'origine, leurs conditions de conservation ne sont plus comparables aux données du fabricant et la responsabilité du pharmacien est alors engagée. D'autre part, des contaminations croisées peuvent être envisagées. A noter que ce type d'automate existe aussi pour la dispensation globale, ils préparent alors des médicaments « en gros » pour des services, il est utilisé par exemple au **CHU** de Grenoble [Bedouch 09].
- Les automates de dispensation « toutes formes » centralisés au niveau de la **PUI** permettent de conditionner des formes orales sèches, formes orales liquides et injectables. Ils permettent la préparation des médicaments par patient et par heure d'administration à partir de leur emballage d'origine.
- Les armoires automatisées sécurisées délocalisées dans les unités de soins. Ces automates permettent la réalisation de la cueillette. Ils sont particulièrement adaptés aux services pour lesquels la dispensation centralisée à la **PUI** est impossible. C'est par exemple le cas des réanimations, des services d'urgences où le délai entre la prescription et l'administration doit être le plus bref possible. Ces armoires sont placées au niveau des unités de soins. Elles possèdent des tiroirs dont l'ouverture est commandée par le système informatique. Les médicaments (formes orales, injectables) sont stockés sous forme unitaire reconditionnée ou sur-conditionnée permettant leur identification. La connexion des armoires à un serveur de la **PUI** permet la gestion en temps réel du stock. L'approvisionnement est sous le contrôle de la **PUI**. L'objectif de ces armoires est d'une part la sécurisation de l'administration - la spécialité pharmaceutique est complètement identifiable - et d'autre part la traçabilité complète de l'administration puisqu'à chaque sortie de médicament de l'armoire, une lecture du code barre est réalisée.

Le choix entre une automatisation centralisée ou décentralisée repose sur des variables comme le coût, le volume, l'activité de l'établissement, l'organisation spatiale des locaux et de l'établissement. Chaque établissement doit choisir la solution la plus adaptée au contexte

de l'hôpital. Généralement, les hôpitaux auront besoin d'imaginer un système mixte pour répondre de manière adaptée à l'ensemble des services hospitaliers [Labrosse 06].

Les photographies de la figure 1.1 montrent les principaux automates « toutes formes » existant aujourd'hui.



FIGURE 1.1 – Exemples de automates « toutes formes » de préparation nominative

A gauche, l'automate de la société MC Kesson² qui est le leader en Amérique du Nord. Au centre, l'automate de la société Swisslog³ et à droite, l'automate de la société Sinteco⁴. Les automates de Swislog et Sinteco sont déjà en place dans plusieurs hôpitaux en France comme par exemple dans les hôpitaux de Lyon et d'Annecy. Un des automates « formes orales sèches » le plus répandu en Amérique du Nord, de la société Mc Kesson, est représenté sur la figure 1.2.



FIGURE 1.2 – Automates « formes orales sèches » PACMED avec des pilules ensachées

Les principaux avantages de l'utilisation des nouvelles technologies sont de permettre une traçabilité (définition 1.4) du médicament et de son utilisation tout au long de son parcours, la baisse du temps infirmier consacré à la gestion des médicaments, une meilleure gestion des stocks et enfin, le développement de la pharmacie clinique (conseil au médecin des pharmaciens).

2. www.mckesson.com (accessible au 13 juillet 2010)

3. www.swisslog.com (accessible au 13 juillet 2010)

4. www.sintecorobotics.com (accessible au 13 juillet 2010)

Définition 1.4 La traçabilité est la possibilité de suivre un produit aux différents stades de sa production, de sa transformation et de sa commercialisation [Pélissier 00].

L'informatisation et l'automatisation du CDM semblent donc inéluctables lorsqu'on envisage la dispensation nominative sur des établissements comme les CHU [Labrosse 06]. De nombreux hôpitaux européens étudient la possibilité d'utiliser des systèmes robotisés pour la dispensation de médicaments [Cairns 01]. Une telle configuration est représentée sur la figure 1.3.

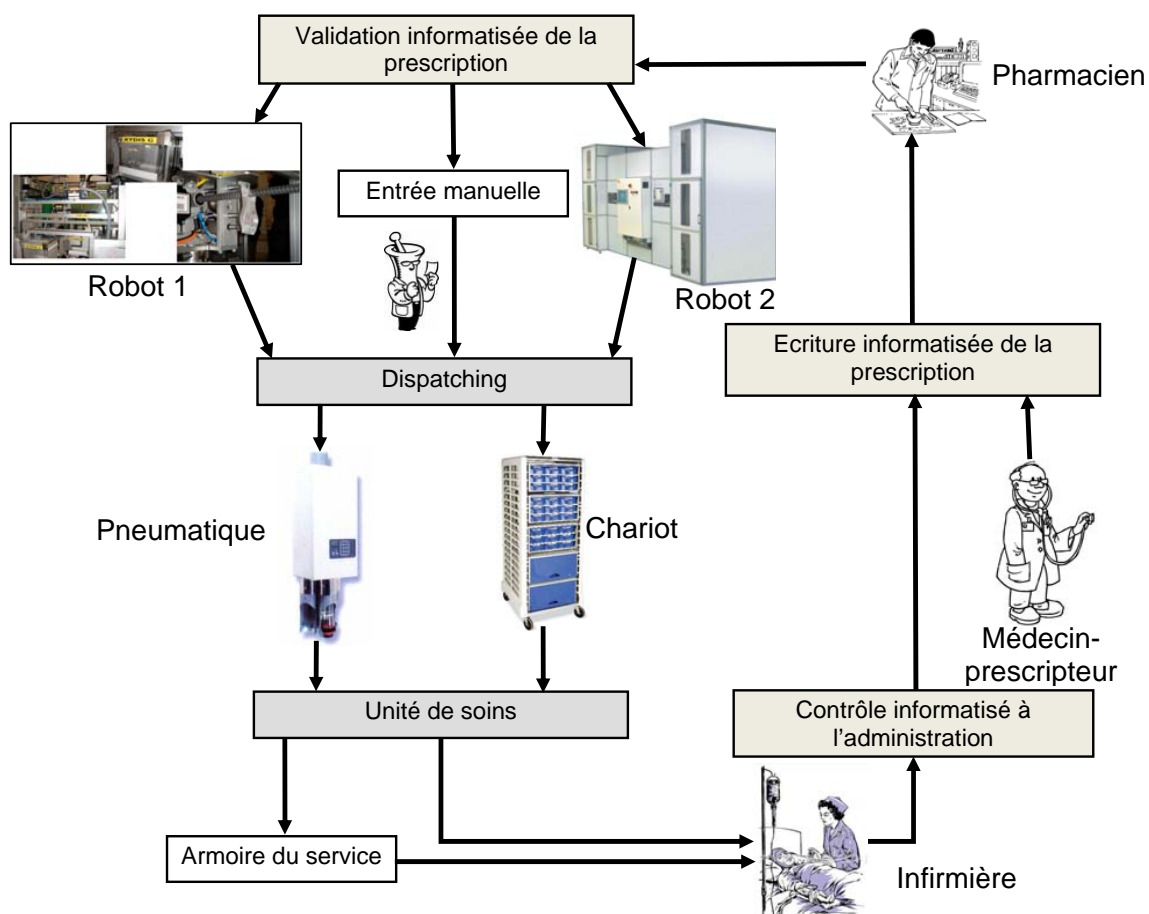


FIGURE 1.3 – Description des entités du circuit du médicament pour une dispensation nominative [Huet 10a]

Elle montre l'ensemble des entités qui peuvent composer le processus étudié pour un fonctionnement avec une dispensation nominative. Plusieurs robots (automates « toutes formes » ou « formes sèches ») et plusieurs préparateurs préparent les produits de la prescription pour un patient donné et pour une durée donnée en utilisant les médicaments qu'ils ont chacun en stock. Ensuite, toutes ces préparations sont collectées et transportées dans les unités de soin par un circuit pneumatique ou par chariots. Le transport par chariots peut être automatisé avec des « tortues ». Nous utilisons un exemple avec deux robots pour cette illustration. Le détail du circuit pneumatique utilisé au CHU de Clermont-Ferrand est donné en annexe A.1. Un système idéal envisagé serait l'utilisation de robots complétée par une armoire automatisée au niveau de la PUI pour les médicaments ne pouvant être stockés dans les robots, puis

une armoire automatisée sécurisée dans l'unité de soins pour les urgences et enfin, pour ce qui ne peut être utilisé dans ces automates, une gestion globalisée en Plein/vide.

1.3 SYNCHRONISATION DES FLUX PHYSIQUES ET INFORMATIONNELS

1.3.1 Synchronisation des flux du CDM

La sécurisation du **CDM** implique une bonne coordination entre les flux matières et informationnels. L'utilisation de nouvelles technologies nous permet d'explorer de nouvelles façons de gérer au mieux cette synchronisation. L'ajout d'un code-barre ou d'une puce électromagnétique **RFID** sur un ensemble constitué de doses préparées d'après une prescription, ensemble que nous appellerons « les doses préparées pour le patient X », permet de rassembler les informations sur cet ensemble et sert de support aux interactions avec d'autres éléments physiques et informationnels du circuit du médicament.

Nous allons voir dans les prochaines sections que nous pouvons comparer un **CDM** avec un système de production, ce qui va nous permettre de récupérer toutes les méthodologies associées à ce domaine.

1.3.2 Interopérabilité avec le Système d'Information Hospitalier

Les flux informationnels sont portés par les logiciels du *Système d'Information Hospitalier* (**SIH**). [Mille 08] présente cinq applications informatiques qui sont souvent en association (module d'un logiciel) :

- Une prescription informatisée,
- Un système d'aide à la décision pour le prescripteur,
- Les systèmes de gestion des automates de dispensation et circuit pneumatique,
- Un système code-barre ou **RFID** pour identifier les médicaments et les patients, ce qui permet d'assurer que le bon médicament est donné au bon patient au bon moment,
- Un système de gestion de la traçabilité de l'administration.

Par exemple, au **CHU** de Clermont-Ferrand, la prescription du médecin sera établie dans le logiciel Horizon Expert Order (HEO) de la société McKesson ; elle sera ensuite transmise au logiciel Pharma de la société Computer Engineering qui permet l'analyse et la validation de l'ordonnance et la préparation des doses. La validation de l'administration des doses au patient par l'infirmier se fera dans le logiciel Crossway de la société McKesson. Ces logiciels devront ainsi interopérer avec les automates de dispensation et le système de transport (circuit pneumatique par exemple). A noter qu'il existe de nombreux autres logiciels pour la gestion pharmaceutique utilisés dans d'autres établissements de santé, comme par exemple le logiciel Génois [Madic 01]. De plus, nous devons ajouter à ces cinq applications le système de gestion économique (GEF) et le système d'information clinique (SIC) correspondant à des services ou produits particuliers. Nous rencontrons donc ici les mêmes problèmes d'interopérabilité que dans le milieu industriel. Une norme française d'interopérabilité a été écrite pour gérer la communication entre les applications informatiques consacrées du **CDM** : la norme PN 13 [Roux 10b].

1.3.3 Interaction avec le flux physique

Nous considérons que le système que nous étudions a la fonction de préparer des boîtes de médicaments qui sont entreposées à la pharmacie [Huet 09c]. Dans chaque boîte, nous

disposons la préparation de la prescription pour un patient pour une période spécifique. La période pour chaque unité de soins peut être par exemple d'un jour (DJIN) ou d'une semaine. Les photos de la figure 1.4 montrent différents exemples de conditionnement, qui seront assemblés dans les boîtes, faits par les robots existants. Nous comparons ces éléments conditionnés à des pièces dans l'industrie.



FIGURE 1.4 – Différentes préparations de robots (à droite SWISSLOG, les autres SINTECO)

Nous voyons donc le circuit du médicament comme un système de production (définition 1.5) et plus précisément un système d'assemblage [Huet 09b, Huet 09c]. La prescription est vue comme une nomenclature dans l'industrie. Chaque ajout de médicament est comparable à l'ajout d'une pièce dans le milieu industriel.

Définition 1.5 *Un système de production met en œuvre un ensemble de processus de transformation qui convertissent des matières premières et/ou des produits semi-finis d'entrée en des produits finis de sortie ayant une valeur ajoutée, et ce en utilisant systématiquement des ressources matérielles et humaines [Leitão 04].*

La table 1.1 montre comment nous pouvons faire l'analogie entre les entités existantes dans un système de préparation de médicament et celles d'un système d'assemblage avec le vocabulaire employé.

Système de production	Circuit du médicament	Ressources utilisées dans le circuit du médicament
Opération d'assemblage/transformation	Ajout de médicaments, modification des doses en préparation	Automates préparateurs, pharmaciens et préparateurs
Opération de contrôle	Contrôle automatique ou par un préparateur	Pharmaciens, infirmières, SIH
Opération de maintenance/stockage	Transport, zone de stockage	Circuit pneumatique, chariots...
Nomenclature	Prescription	Médecins

TABLE 1.1 – Description de l'analogie avec les ressources utilisées [Huet 09b, Huet 09c]

D'autres auteurs ont déjà fait une analogie avec le milieu industriel, par exemple [Aloui 07] et [Ghorfi 10] qui utilisent les concepts du milieu industriel dans la gestion des risques en l'appliquant au CDM.

Comme exemple de processus d'assemblage, nous utilisons le processus d'assemblage d'une carte électronique (figure 1.5).

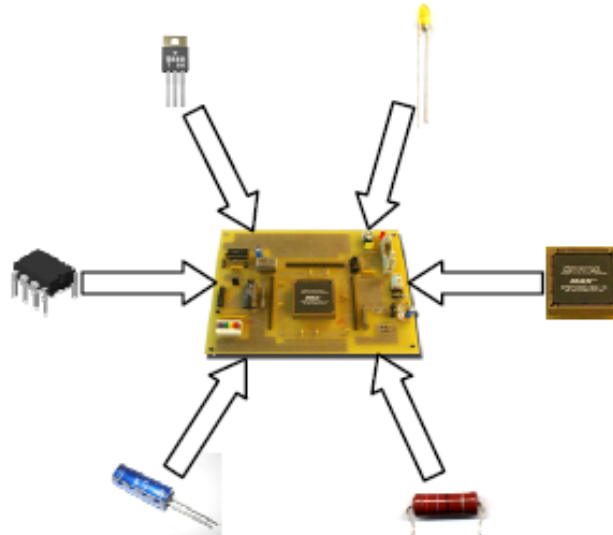


FIGURE 1.5 – Exemple de système d'assemblage : une carte électronique

Six différents composants peuvent être assemblés de différentes façons pour obtenir différents produits (cartes électroniques). Avec différents assemblages, nous avons différents produits avec différentes pièces. Dans le circuit du médicament, nous pourrions considérer le produit final comme la préparation des doses pour un patient. En effet, une pièce correspond à un médicament, bien sûr dans la réalité, dans une PUI, beaucoup de médicaments sont stockés et sont répartis entre plusieurs robots ou gérés par les préparateurs. En fait, la prescription joue le rôle que joue une nomenclature dans un système d'assemblage.

1.4 ANALOGIE ENTRE LE CDM ET UN FMS

Nous cherchons par conséquent dans le milieu industriel les domaines qui pourraient nous intéresser, sachant que nous essayons d'avoir le maximum de robustesse, de flexibilité et de modularité. On peut trouver de nombreux articles qui font référence aux problèmes sur les *Flexible Manufacturing System - Système manufacturier flexible (FMS)* [Cardin 08, Sallez 04, Trentesaux 08]. Ce paradigme a été créé dans le but de faire face à une grande personnalisation dans les demandes des clients. Par conséquent, les systèmes de contrôle industriels ont besoin de développer leurs capacités à s'adapter à cette variation de la demande en terme de spécification de produits ou de changement intrinsèque du système. Pour ce faire, un système de contrôle flexible doit prendre en compte un maximum de changements de politique de contrôle et doit être conçu pour traiter les variations de routage de la production industrielle, ainsi que les redondances fonctionnelles entre les machines. Les progrès dans l'utilisation des technologies de l'information dans les systèmes de production donnent aux industriels une opportunité de promouvoir la personnalisation des produits. Cette personnalisation des

produits avec l'utilisation de nouvelles technologies d'identification comme la **RFID** ont fait émerger un nouveau paradigme : le *Système Contrôlé par le Produit* (**SCP**). Dans la littérature sur les **FMS** et sur les **SCP** on peut trouver plusieurs exemples qui concernent les systèmes d'assemblage [[Cardin 08](#), [Fontanili 99](#), [Heverhagen 01](#), [Pétin 07](#), [Sallez 04](#), [Trentesaux 08](#)].

Notre approche se concentre sur la conception d'un système de contrôle distribué piloté par le produit, basée sur la coopération entre [[Gouyon 07](#)] :

- Les contrôleurs du produit qui dirigent le routage en accord avec la programmation dans une liste d'opérations que le produit doit suivre. Ces contrôleurs sont spécifiques pour chaque occurrence de produit dans le but de prendre en compte leur personnalisation.
- Les contrôleurs des ressources qui assurent une exécution correcte du transport et des opérations de transformations et fournissent les contrôleurs avec des rapports exacts. Le contrôle de la flexibilité connecte les paramètres de réglage de l'objet fonctionnel qui coordonne et contrôle les opérations élémentaires, ou chargent les politiques de contrôle spécifiques embarquées dans le produit.

Nous nous sommes alors intéressés aux différents types de système d'assemblage en les comparant et cherchant le plus proche d'un **CDM**. Dans la littérature nous avons trouvé de nombreux systèmes d'assemblage qui lui sont similaires. Nous en avons retenu trois types :

- **En série** comme dans [[Fontanili 99](#)], représenté sur la figure 1.6,
- **Avec dérivation** comme dans [[Cardin 08](#), [Pétin 07](#), [Sallez 04](#), [Trentesaux 08](#)], représenté sur la figure 1.7,
- **Sans dérivation** comme dans [[Heverhagen 01](#)], représenté sur la figure 1.8,

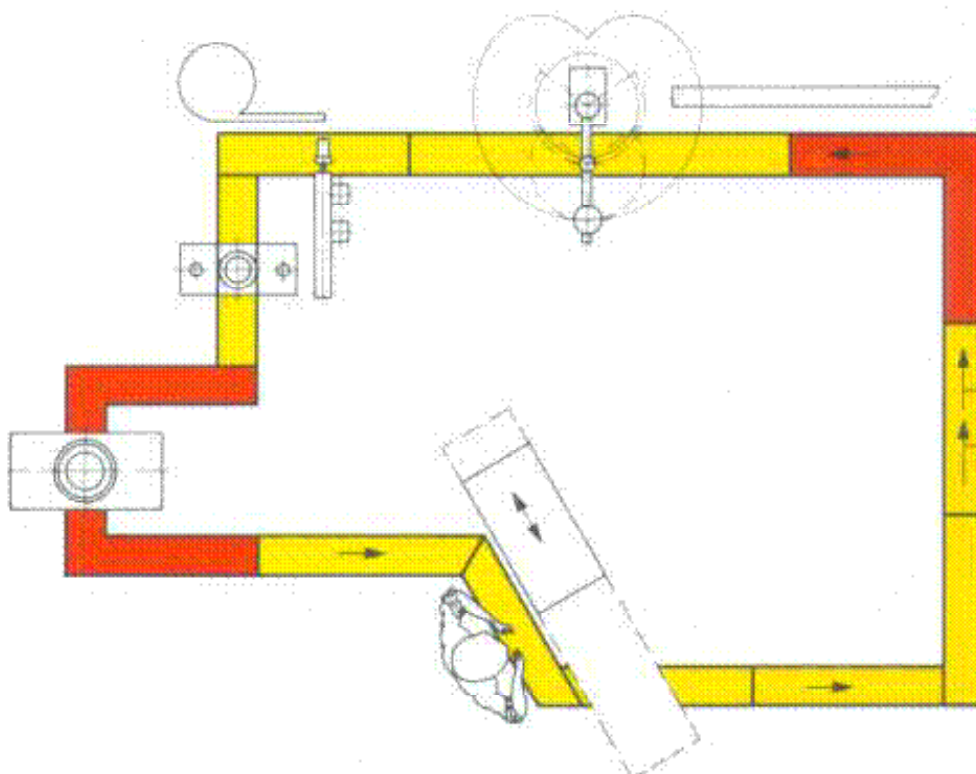


FIGURE 1.6 – Exemple d'un système d'assemblage en série [[Fontanili 99](#)]

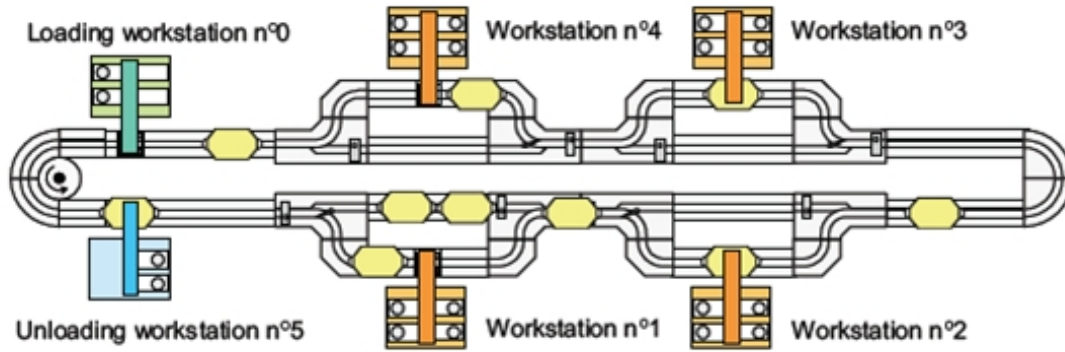


FIGURE 1.7 – Exemple d’un système d’assemblage avec dérivation [Pétin 07]

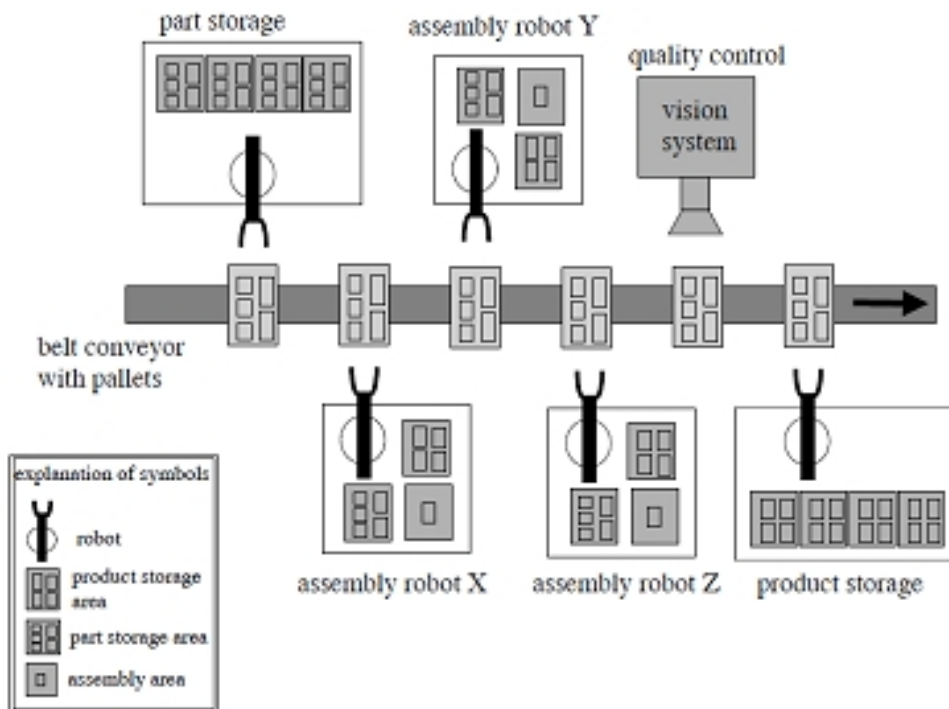


FIGURE 1.8 – Exemple d’un système d’assemblage sans dérivation [Heverhagen 01]

Nous considérons comme un système d'assemblage avec dérivation, un système où le produit doit sortir du chemin principal pour aller effectuer une opération. Il est différent d'un système d'assemblage sans dérivation où les postes de travail sont en série, c'est-à-dire que le produit est pris en charge sans avoir à sortir du chemin principal. Le cas le plus simple est en série : les stations de travail traitent le produit directement sur le convoyeur.

Le choix de l'architecture d'assemblage doit être fait en collaboration avec les pharmaciens.

1.5 CONCLUSION

Nous avons expliqué dans ce premier chapitre le fonctionnement du circuit du médicament. Nous avons montré que les problématiques actuelles de ce domaine sont proches de celles débattues actuellement dans des travaux de recherches sur les concepts tels que le **FMS** et plus précisément sur les **SCP**. Nous avons expliqué que le fonctionnement le plus sécurisé pour le circuit du médicament est le fonctionnement nominative (**DJIN**). C'est avec la **DJIN** que le concept **SCP** prend tout son sens avec le contrôle de la préparation des doses pour un patient à partir de prescriptions contrôlées par puce **RFID** ou code à barre tout au long de son parcours. Notre objectif est de permettre la réingénierie d'un **CDM** en partant de l'organisation actuelle pour obtenir au final une organisation plus sécurisée et plus efficace. Nous avons retenu l'explication de la réingénierie proposée par [Chikofsky 90] que nous reprenons dans la définition 1.6.

Définition 1.6 *La réingénierie d'un système est l'examen et la modification d'un système afin de le reconstituer sous une nouvelle forme et d'implanter cette nouvelle forme [Chikofsky 90].*

Nous avons montré que le **CDM** est un système de production. Or, pour les systèmes de production, beaucoup de méthodologies existent et ont fait leur preuve. Par conséquent, dans un premier temps, nous chercherons à trouver une méthodologie pour permettre la réingénierie d'un système production pour le rendre plus flexible. Par la suite, nous tenterons de l'appliquer au **CDM**. Malheureusement, nous verrons qu'il y a des manques que nous comblerons en apportant une nouvelle méthodologie. Le chapitre suivant détaillera la notion de **SCP**.

2

LE CONTRÔLE PAR LE PRODUIT DES SYSTÈMES DE PRODUCTION

RÉSUMÉ DU CHAPITRE 2

Le chapitre 1 a permis de définir et de détailler le fonctionnement du circuit du médicament. Nous avons aussi expliqué que le circuit du médicament peut être similaire à un système de production flexible et plus précisément à un système contrôlé par le produit. Ce type de système utilise des approches de pilotage décentralisées.

Dans ce chapitre, nous détaillons tout d'abord les différentes structures de prise de décision et leurs problématiques : hiérarchique, hétérarchique et hybride. Une structure complètement décentralisée (hétérarchique) a un inconvénient majeur : un manque de cohérence globale. Il faut donc garder un minimum d'éléments centralisés : la structure hybride nous paraît alors adaptée. La prise de décision hybride peut être plus facilement pilotée par le concept de « holon ». Un holon est un composant ayant une partie informationnelle et physique et pouvant être divisé en autres holons. Nous exposons alors ce concept et les différentes architectures qui s'y rapportent. Dans une entreprise et dans le milieu hospitalier, il y a beaucoup de systèmes d'information qui doivent coopérer entre eux ; nous avons des problèmes d'interopérabilité. La notion d'interopérabilité entre systèmes d'information est présentée ainsi que les différentes façons de la contrôler. Nous faisons alors le lien entre cette notion et celle de système contrôlé par le produit avec le paradigme d'interopérabilité contrôlé par le produit. Grâce au concept de système contrôlé par le produit, nous pouvons donc piloter un système de production à prise de décision hybride et ses différents systèmes d'information (interopérabilité).

SOMMAIRE

2.1	INTRODUCTION	21
2.2	LES DIFFÉRENTES APPROCHES DE PILOTAGE	21
2.2.1	Approche hiérarchique	21
2.2.2	Approche hétérarchique	22
2.2.3	Approche hybride	23
2.3	LE CONCEPT D'HOLON ET DE SYSTÈME MANUFACTURIER HOLONIQUE	24
2.3.1	PROSA	25
2.3.2	Autres architectures de HMS	26
2.3.3	Différence entre une approche basée sur les agents et une approche holonique	27
2.4	LE BESOIN DE PRENDRE EN COMPTE L'INTEROPÉRABILITÉ DANS UNE ENTREPRISE	28
2.4.1	Les différents niveaux de décision dans une entreprise	28
2.4.2	La notion de Système de Systèmes et le Système d'Information Hospitalier	31
2.4.3	Les modèles d'interopérabilité	32
2.4.3.1	Le modèle LISI	33
2.4.3.2	Le modèle LCIM	34
2.5	CONCLUSION	35

2.1 INTRODUCTION

Le contrôle par le produit va dans le sens de la personnalisation de masse (ou « mass customization » en anglais) [Silveira 01]. La personnalisation de masse se rapporte à l'aptitude à fournir des produits ou des services personnalisés dans des quantités importantes, grâce à des processus flexibles, et à des coûts raisonnables [Silveira 01]. Cette place importante donnée au produit [Gouyon 07] a conduit au sein de la communauté *Intelligent Manufacturing Systems* (IMS) à la définition et à l'expérimentation des *Holonic Manufacturing System* (HMS) [Deen 03] qui ont pour but de conférer au produit un rôle actif.

L'utilisation des technologies RFID conduit donc à une modification des pratiques, en conférant aux produits des fonctions nouvelles. Des classifications plus générales de ces fonctions, ne dépendant pas d'une technologie particulière, ont été proposées. Du point de vue des services offerts par l'objet dans le cadre d'une intelligence ambiante, on peut distinguer [Cea Ramirez 06] :

- L'objet porteur de données,
- L'objet pointeur vers un système d'information,
- L'objet fournisseur et demandeur de services,
- L'objet sensitif enrichi avec des capteurs et des actionneurs.

Ces nouvelles technologies font émerger de nouveaux concepts pour permettre un pilotage plus flexible des systèmes de production. Nous proposons dans ce chapitre de rappeler les différentes approches de pilotage, de détailler le concept d'holon et de HMS, d'expliquer les différents niveaux de décisions dans une entreprise et enfin de montrer le besoin de prendre en compte l'interopérabilité dans le *Système d'Information d'Entreprise* (SIE) qui en découle. Nous concluons alors ce chapitre en résumant nos problématiques, dans les systèmes de production actuels et plus particulièrement sur le *Circuit Du Médicament* (CDM) d'un hôpital.

2.2 LES DIFFÉRENTES APPROCHES DE PILOTAGE

L'accroissement des besoins industriels d'agilité et de traçabilité, ainsi que l'apparition de nouvelles possibilités techniques ou fondamentales, conduisent à la proposition de systèmes de pilotage distribués avec une structure hétérarchique, qui s'opposent à des méthodes de gestion plus classiques, centralisées [Pannequin 07] avec une structure plus hiérarchique (figure 2.1).

2.2.1 Approche hiérarchique

Dans cette approche très centralisée (figure 2.1a), une hiérarchie forte existe entre toutes les composantes du système qui s'appuie sur un flot descendant de commandes et un flot remontant d'informations. Ainsi, un niveau supérieur définit les contraintes et objectifs à atteindre par le niveau suivant. De ce fait, un problème de coordination entre les différents niveaux apparaît, puisqu'il est très fréquent que les problèmes de chaque niveau soient interdépendants. Il faut donc instaurer des mécanismes assurant la cohérence des décisions prises à chaque niveau [Cardin 07].

Le principal avantage d'une telle structure est qu'elle donne une vision globale du système de production. Cette vision globale permet non seulement la définition claire d'objectifs globaux mais elle permet aussi de prendre des décisions cohérentes à chaque niveau pour atteindre ces objectifs globaux. Par contre, les évolutions de la demande (multiplication des produits et des variantes), et la complexification croissante des systèmes de production due

à la mondialisation font qu'aujourd'hui cette vision hiérarchique pose de plus en plus de problèmes. Les avantages d'une telle approche sont nombreux (optimisation des objectifs globaux ou fiabilité des données par exemple) mais ce qui nous intéresse plus particulièrement concerne sa structure même. En effet, les chemins de remontée d'informations sont particulièrement bien définis, ce qui permet de ne communiquer qu'avec le niveau supérieur pour obtenir l'ensemble des informations qui nous intéressent. Toutefois, la rigidité de l'architecture a tendance à diminuer fortement la réactivité du système. Ce pilotage très hiérarchisé est en effet beaucoup trop peu réactif pour répondre aux contraintes multiples des productions actuelles [Cardin 07].

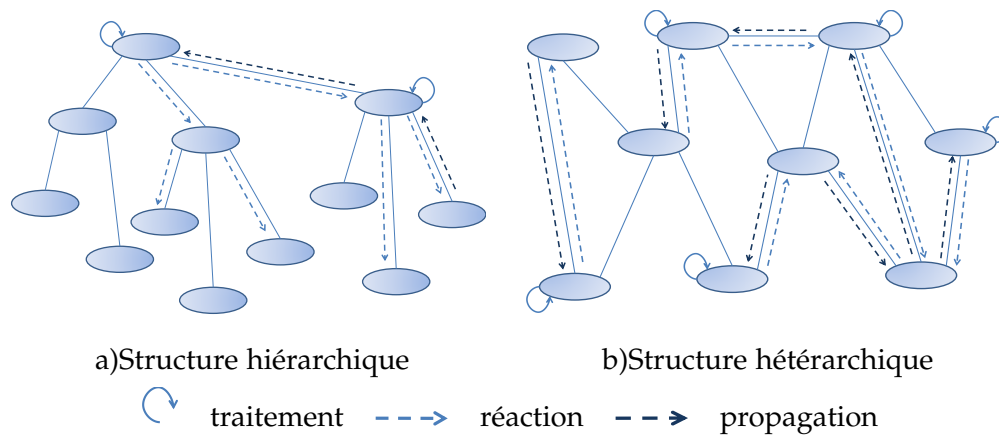


FIGURE 2.1 – Exemples de structures de pilotage [Cardin 07]

2.2.2 Approche hétérarchique

Le substantif hétérarchie (en anglais, heterarchy) et son adjectif hétérarchique (en anglais, heterarchical) constituent des néologismes à la fois en français et en anglais. Ce terme a été formé à partir de deux mots grecs : heteros (autre) et Arkhein (commander), signifiant ainsi à l'origine « commandement par les autres » [Trentesaux 02]. L'hétérarchie renvoie à l'idée d'acteurs différents qui assument en collégialité la coordination d'une action collective donnée et s'oppose par essence au terme hiérarchie (figure 2.1b). L'information et la décision sont donc distribuées, ce qui impose d'instaurer des mécanismes de coordination : chaque proposition est transmise à tous les agents concernés qui peuvent l'accepter ou la remettre en cause.

A noter que [Pujo 09] utilise le terme de « isoarchie » pour qualifier un système qui est à la fois décentralisé avec un seul niveau de décision et auto-organisé en temps réel. Ce terme a été formé à partir de deux mots grecs : isos (égale) et Arkhein (commander). Il signifie que l'autorité est la même pour tous les centres de décision, c'est-à-dire avec une totale absence de hiérarchie. Une telle structure doit être aussi fonctionnelle dans le temps.

L'utilisation de techniques d'intelligence artificielle dans l'approche distribuée lui confère des qualités d'adaptabilité à des conditions changeantes et incertaines [Pannequin 07]. La localisation des prises de décision au plus près du terrain lui donne une grande réactivité aux aléas. De plus, l'utilisation d'une approche objet facilite les tâches de création et de maintenance d'un modèle global, puisque les propriétés globales du système émergent d'un ensemble de comportements simples. La simplicité même du fonctionnement interne des centres de décision leur confère une grande réutilisabilité.

Depuis le milieu des années 1990, des structures hétérarchiques se sont développées où l'orientation des relations entre les différents éléments constitutifs du système n'est pas définie a priori. Ces éléments sont capables d'agir et de communiquer, ainsi que d'accéder aux ressources ou d'interagir : on les appelle des agents. Il n'existe pas de définition unique et consensuelle pour un agent ou un système multi-agent. Néanmoins, les définitions 2.1 et 2.2 nous semble les plus adaptées [Darmoul 10].

Définition 2.1 *Un agent est un système logiciel qui communique et coopère avec d'autres systèmes logiciels pour résoudre un problème complexe dont la résolution dépasse la capacité individuelle de chacun des systèmes logiciels pris à part [Shen 06].*

Définition 2.2 *Un système multi-agent est un système dans lequel l'abstraction clef utilisée est celle d'agent [Shen 06].*

La prise de décision distribuée est couramment critiquée par rapport à son comportement essentiellement réactif, pouvant conduire à des décisions qui ne sont pas globalement optimales, dans la mesure où seules les conséquences immédiates dans le temps ou l'espace sont prises en compte. L'approche distribuée paraît de ce point de vue relativement inadaptée aux décisions ayant un impact à moyen ou long terme [Pannequin 07]. L'avantage d'une telle structure est bien évidemment sa robustesse et son adaptabilité [Sallez 04] : l'ajout d'un agent ou la suppression temporaire pour cause de panne par exemple sont très faciles à mettre en œuvre dans une telle structure [Cardin 07].

2.2.3 Approche hybride

D'après les apports respectifs des approches centralisées et distribuées de pilotage de la production, nous pouvons faire apparaître deux champs d'application préférentiels [Pannequin 07]. D'abord, l'approche centralisée, de par sa prise en compte globale du système, et sa capacité à programmer les activités futures, permet de remplir un ensemble de fonctions que l'on peut désigner comme la gestion logistique, ou la gestion de la chaîne logistique. C'est l'ensemble des activités de planification et de mise en œuvre, permettant de coordonner les activités de chaque acteur de la chaîne logistique. Ces activités incluent l'approvisionnement, la production, la distribution, la vente.

Ensuite, l'approche distribuée, de par ses capacités de réactivité et d'adaptation à des conditions changeantes, correspond aux besoins du domaine de l'exécution de la production. L'exécution (ou pilotage) de la production, est l'ensemble des activités de coordination des équipements industriels, des matières, de l'énergie et des ressources humaines, afin de convertir les matières premières en produits finis. Les systèmes de pilotage intelligents sont les plus à même de répondre aux besoins croissants de gestion de la variété, de robustesse face à un environnement incertain et de reconfigurabilité, de par leurs caractéristiques d'auto-organisation et d'adaptativité. De plus, le pilotage de la production est par nature un problème d'évolution, puisqu'il s'agit de maintenir le système en condition dans un environnement perturbé et incertain. Un système de production idéal comporterait donc les deux approches, centralisées et distribuées, de prise de décision. Celles-ci, grâce à un système assurant un couplage et un bon équilibre, conjugueraient leurs avantages. Toutefois, trouver le bon équilibre entre les fonctions centralisées et distribuées n'est pas trivial [Bongaerts 00]. La réalisation de tels

systèmes de pilotage hybrides (au sens où ils combinent l'approche centralisée et distribuée) a été l'objectif de la communauté des *Holonic Manufacturing System (HMS)* depuis une dizaine d'années (figure 2.2). Cette communauté a proposé par la suite de mettre le produit au cœur du système pour permettre de faire le lien entre ces deux approches.

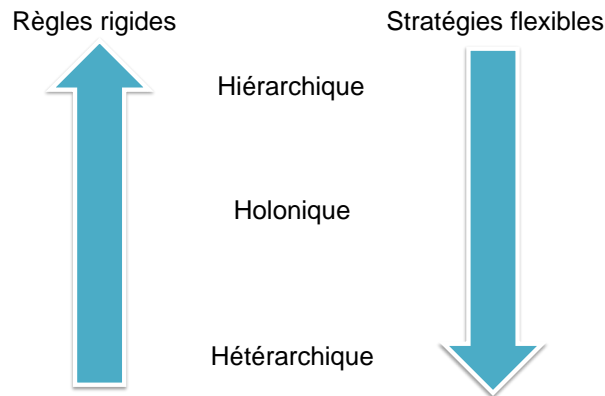


FIGURE 2.2 – Balance entre différents types de pilotage suivant la flexibilité d'après [Bongaerts 00]

2.3 LE CONCEPT D'HOLON ET DE SYSTÈME MANUFACTURIER HOLONIQUE

[Koestler 67] décrit un « holon » comme une entité autonome faisant partie d'un « tout », c'est-à-dire capable de communiquer et coopérer avec son environnement. Dans [Brussel 98], les auteurs présentent un holon comme une entité autonome composée d'une partie informationnelle et d'une partie physique (figure 2.3).

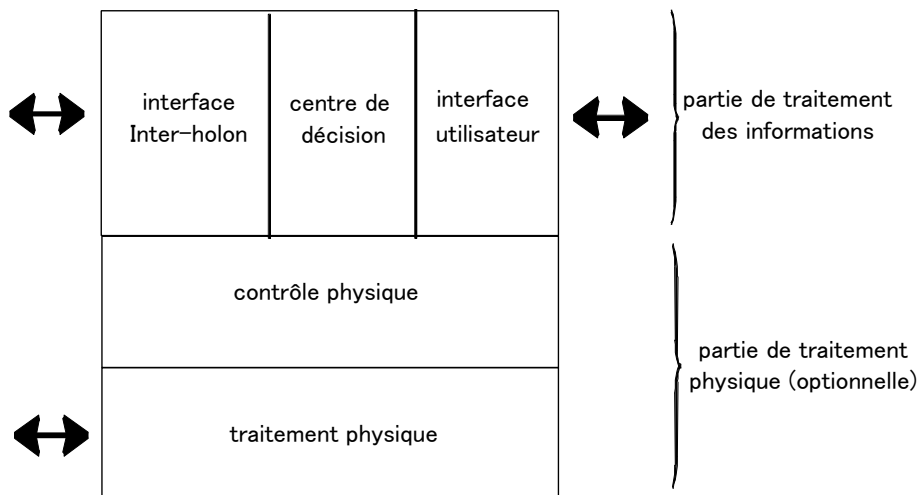


FIGURE 2.3 – Architecture générale d'un holon [McFarlane 00]

L'architecture de référence la plus répandue est l'architecture **PROSA** pour modéliser un *Holonic Manufacturing System (HMS)* : elle définit principalement une architecture inter-holon, identifiant les types d'holons, leurs responsabilités et la façon dont ils interagissent. Elle est constituée de trois types d'holons de base : holon-ordre, holon-produit et holon-ressource, dans lesquels les données et les traitements de la production sont distribués. Il existe aussi

d'autres types d'architecture tel que **ADACOR** ou **HCBA**. Ces concepts permettent de structurer un système distribué afin de gérer le pilotage des systèmes de production (biens et services).

2.3.1 PROSA

L'architecture de référence *Product-Resource-Order-Staff Architecture* (**PROSA**) consiste en quatre types d'holons composés de trois types d'holons basiques et d'un type d'holon optionnel. Les trois types d'holons basiques sont les suivants : Product-Holon, Resource-Holon et Order-Holon [Brussel 98, Valckenaers 98], l'holon optionnel est le Staff-Holon. Nous traduirons en français par Holon-Produit, Holon-Ressource, Holon-Ordre et Holon-Staff. Les trois holons basiques sont définis ainsi [Brussel 98] :

- Un **Holon-Produit** contient la connaissance nécessaire relative au produit et au procédé pour assurer la fabrication d'un produit avec une qualité correcte. Il a le rôle d'un serveur d'informations pour les autres holons du système. Il fournit des informations consistantes et à jour sur le cycle de vie du produit, les spécifications utilisateurs, sa gamme de fabrication et sa nomenclature,
- Un **Holon-Ressource** est constitué d'une partie physique (une ressource de production du HMS), et d'une partie de traitement de l'information qui contrôle cette ressource. Il offre une capacité de production aux holons alentours. Il contient des méthodes pour allouer des ressources de production, ainsi que la connaissance et les procédures pour organiser, utiliser et contrôler les ressources pour conduire la production. Un Holon-Ressource est une abstraction pour des moyens de production tels que des machines, des outils, stocks de matières. . .
- Un **Holon-Ordre** représente un ordre de fabrication. C'est une entité active responsable de l'exécution correcte et à temps d'un travail. Il se charge explicitement des informations et du traitement des informations concernant un travail.

Un système composé de relations entre ces différents types d'holon est appelé un *Holonic Manufacturing System* (**HMS**). Leurs relations sont expliquées par la figure 2.4. Leurs relations modélisées en langage **UML** sont présentées sur la figure 2.5.

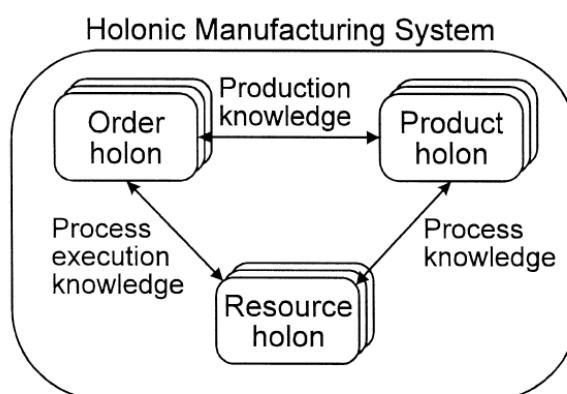


FIGURE 2.4 – Bloc de construction basique d'un HMS et leurs relations [Brussel 98, Valckenaers 98]

Un **Holon-Staff** [Mařík 02] est une unité optionnelle qui permet la coordination entre les holons et d'assurer que l'objectif général de la production est atteint. Ce type d'holon est un compromis entre une vision purement hétérarchique du contrôle par les services fournis et

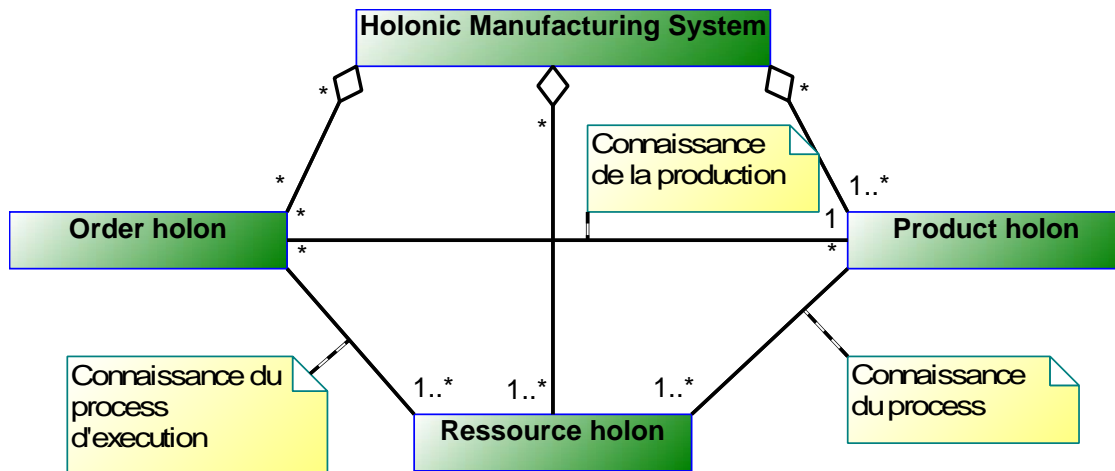


FIGURE 2.5 – Un HMS et ses holons de base avec un diagramme de classe [Valckenaers 98]

une vision plus hiérarchique. Les conseils donnés par les holon-staff peuvent être rejetés par les holons basiques.

[Gouyon 04] propose une adaptation de ces définitions pour permettre le contrôle par le produit. En effet, ces définitions ne correspondent pas directement aux définitions de produits, ressources et ordres qui sont classiquement utilisées en ingénierie et en fabrication. En ce sens, il explique que la définition donnée dans PROSA de l'holon-produit correspond au savoir-faire nécessaire à une classe de produits, et l'holon-ordre, à une instance d'une de ces classes de produits. Il considère que l'holon-produit contient les informations nécessaires à sa fabrication (comme dans PROSA), mais en plus qu'il est unique, et qu'il contrôle sa propre fabrication (comme le fait l'holon-ordre dans PROSA). L'holon-ordre correspond dans ce cas à l'ordre de fabrication d'un ou de plusieurs produits. Dans le cas où cet ordre ne concernerait qu'un produit, il correspondrait en quelque sorte au premier état du produit, un état « virtuel ».

2.3.2 Autres architectures de HMS

Il existe d'autres architectures de référence pour les HMS. Celles qui nous semblent les plus utilisées sont les architectures *ADaptive holonic COntrol aRchitecture for distributed manufacturing systems* (ADACOR), *Holonnic Component-Based Approach* (HCBA) et *Product, Resource, Order and Simulation Isoarchic Structure* (PROSIS).

L'architecture ADACOR [Leitão 04] est définie par quatre classes d'holons : les « Product-Holon » (PH), les « Task-Holon » (TH), les « Operational-Holon » (OH) et les « Supervisor-Holon » (SH). Les PH, TH et OH sont très proches des Holon-Produit, Holon-Ordre et Holon-Ressource définis dans PROSA. Par contre, les SH présentent des caractéristiques que l'on ne retrouve pas dans les Holon-Staff. Le SH introduit une coordination et une optimisation globale avec un pilotage décentralisé et est responsable de la formation et de la coordination d'un groupe de holons [Leitão 06]. Cette architecture permet de gérer des pannes machines, des retards d'opération, des annulations ou des modifications d'ordres et des ordres urgents.

Elle est basée sur les ressources et leurs coordinations, ce qui permet une bonne robustesse (capacité de gérer perturbations, dégradations des performances, tolérance aux fautes) mais une perte importante de la flexibilité.

[Chirn 00] avec *Holonic Component-Based Approach* (HCBA) propose d'utiliser l'approche par composant dans la conception de HMS. Ils présentent une architecture spécifique pour le pilotage de systèmes holoniques en utilisant les holons produit et ressource de PROSA qu'ils considèrent comme des composants. Le but est de permettre de reconfigurer facilement un système manufacturier. Le système holonique est conçu en associant ces deux types d'holons pour la conception de structures à plusieurs niveaux composées elles-mêmes de produits et de ressources.

Le concept de PROSIS hérite du concept de base de PROSA. Il est caractérisé par une décentralisation totale et une auto-organisation (structure isoarchique). Toutes les décisions sont prises par les différents holons du système de pilotage du système de production. Cette approche est basée sur une recherche de l'anticipation à court terme en utilisant des mécanismes de communication entre holons [Pujo 09]. Les inconvénients et les avantages que l'on peut y trouver sont par conséquent les mêmes que ceux exprimés plus haut pour une approche hétérarchique.

2.3.3 Différence entre une approche basée sur les agents et une approche holonique

Les paradigmes des manufactures basées sur les agents et les manufactures holoniques sont développés sur les mêmes principes fondamentaux d'autonomie et de coopération, explorant la distribution et la décentralisation des entités et des fonctions [Leitão 04].

En terme d'origine, les agents ont leurs racines dans l'informatique (intelligence artificielle) et les holons, dans le domaine du génie informatique pour la productique, en se concentrant sur les problèmes associés aux systèmes manufacturiers flexibles. En terme conceptuel, le holon est un concept, et l'agent est à la fois un concept et une technologie associée; il est possible d'implémenter le concept d'holon sur des systèmes manufacturiers en utilisant la technologie agent [Mařík 02].

En terme d'intégration des ressources physiques, les agents, normalement, représentent des composants logiciels et leur centre d'intérêt n'est pas l'intégration de ressources physiques. Dans le monde manufacturier ce dernier point est critique et le concept d'holon supporte l'intégration de ressources physiques, basée sur le fait que les holons comportent des composants logiques et physiques. En terme d'intégration humaine, l'interface humaine est automatiquement embarquée dans chaque holon, tandis que dans l'approche agent, l'interface humaine est représentée par des agents séparés [Mařík 02].

En terme de contrôle en temps réel, un agent ne peut pas garantir des contraintes temps réel, tandis que les holons doivent rencontrer des contraintes temps réel difficiles qui sont requises pour réussir des opérations fiables. Enfin, un holon peut représenter simultanément un tout et une part d'un tout, l'holon peut être composé par plusieurs sous-niveaux d'holon; un agent ne supporte pas cette caractéristique [Brussel 98].

Ces différences tendent à être réduites parce qu'actuellement, les communautés holoniques et agents convergent rapidement. Cette convergence est reflétée dans les efforts faits par les organismes de normes comme l'IEEE d'introduire des directives et des spécifications qui supportent les exigences holoniques dans les normes pour les agents [Mařík 02].

2.4 LE BESOIN DE PRENDRE EN COMPTE L'INTEROPÉRABILITÉ DANS UNE ENTREPRISE

Nous avons vu dans les sections précédentes les apports et les limites des deux modes de prise de décision. Les pratiques industrielles ont conduit les entreprises à implanter des systèmes de niveau business étant de fait centralisés. Cependant, elles ont implanté des systèmes de niveau process pouvant être distribués. La nécessaire coexistence de ces deux types de systèmes pose la problématique de leur interaction entre les différents niveaux de logiciels utilisés. Un objectif à atteindre est justement de permettre l'interopérabilité (définition 2.3) entre ces différents logiciels pour garantir une cohérence globale.

Définition 2.3 *L'interopérabilité est la capacité d'un système ou d'un produit à travailler avec d'autres systèmes ou produits sans un effort spécial de la part du client. L'interopérabilité est rendue possible par l'implantation de standards [IEEE 00].*

2.4.1 Les différents niveaux de décision dans une entreprise

Le cadre CIM (Computer Integrated Manufacturing) décrit 4 niveaux de décision dans une entreprise. Un niveau supérieur décide ce qu'un niveau inférieur exécute (figure 2.6).

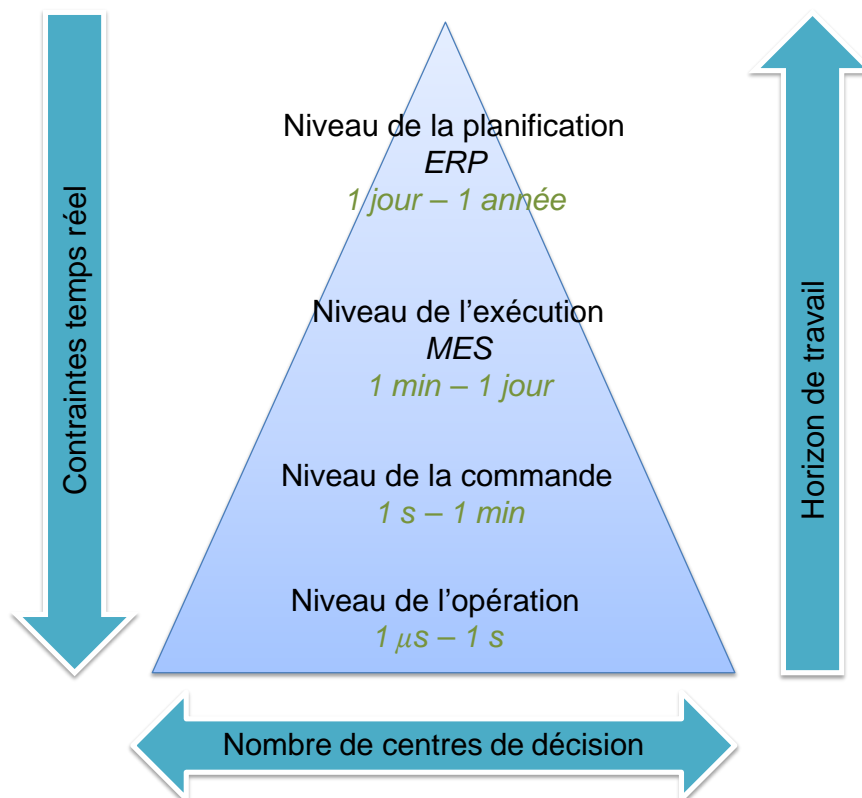


FIGURE 2.6 – Les différents niveaux de décision d'une entreprise

Le niveau supérieur (la planification, niveau 3) contient toutes les applications permettant une vision à long terme de l'évolution du système. Diverses décompositions sont encore faites à l'intérieur du niveau. Parmi les applications rencontrées, nous pourrions citer [Cardin 07] :

- Les ERP : Enterprise Resources Planning. Système d'information intégré d'entreprise adressant l'ensemble (transactions et exécution) des fonctions classiques d'une entreprise, comptabilité, commercial et ventes, production et gestion des matières, administration des ventes, maintenance. Les différents modules et fonctions sont intégrés autour d'un référentiel de processus unique. L'organisation des traitements et des données de tels systèmes répond à des préoccupations de découpage analytique par fonction, de suivi des performances financières et d'exécution, de suivi des principales (ex : SAP, Oracle, Adonix, etc.).
- Les SCM : Supply Chain Management. C'est le processus de planification, d'implémentation et de contrôle de la supply chain. Le SCM englobe les transports, les stockages et les transformations des matières premières, des encours de production, et des produits finis du point d'origine des produits à leur point de consommation. Le principe de base de la SCM est qu'on obtient un résultat moins bon en optimisant chaque élément de la chaîne, plutôt qu'en optimisant la chaîne globale (la somme des optimums de chaque élément de la chaîne est inférieure à l'optimum global).
- Les APS : Advanced Planning and Scheduling. L'APS est une application destinée à la planification. En fonction de la demande, elle permet d'analyser la capacité des ressources et les contraintes afin de proposer un horaire détaillé et adaptable pour une production optimale.

Le niveau en dessous (l'exécution, niveau 2), contient les applications permettant la gestion des produits de leur entrée dans le système à leur sortie. Ce niveau est de plus en plus composé du seul *Manufacturing Execution System* (MES) [Huang 02]. Un MES peut être défini de plusieurs façons :

Définition 2.4 *Un MES est un ensemble de composants matériels et logiciels permettant la gestion et l'optimisation des activités de production du lancement de la commande aux produits finis. Tout en maintenant des données précises et à jour, le MES guide, initie, répond et rapporte les événements de l'usine au moment où ils se produisent. Un MES fournit des informations critiques aux modules d'aide à la décision à travers l'entreprise [Barkmeyer 99].*

On peut compléter cette définition par celle donnée par le « MES Portal »¹ (définition 2.5).

Définition 2.5 *Un MES est composé d'activités et de systèmes permettant de piloter l'ensemble des productions en temps réel et de restituer toutes les conditions dans lesquelles elles se sont déroulées.*

Le MES vient compléter le *Système d'Information d'Entreprise* (SIE) en assurant des fonctions de contrôle/commande qui permettent le pilotage en temps réel des ateliers de fabrication. Une norme américaine et internationale a été mise en place afin de donner les lignes directrices d'un tel type de progiciel, il s'agit de la norme ANSI/ISA-95 issue principalement des travaux du MESA², association américaine à but non lucratif. Cette norme définit entre autres les fonctionnalités que doit remplir un MES ; elles sont au nombre de onze [Cardin 07] :

1. <http://www.mesportal.org> (accessible au 21 octobre 2010)

2. <http://www.mesa.org> (accessible au 21 octobre 2010)

1. Gestion des ressources : Cette fonction suit le statut des ressources et met à jour un historique détaillé.
2. Ordonnancement : Cette fonction fournit le jalonnement des activités indépendantes basé sur des priorités, attributs, caractéristiques et/ou gammes associées aux produits concernés pour respecter les objectifs de performance définis par l'utilisateur.
3. Cheminement des produits et des lots : Cette fonction dirige le flux de produits selon le plan de production et l'ordonnancement détaillé.
4. Gestion des documents : Cette fonction contrôle, gère et restitue diverses informations relatives aux produits, dont les gammes, les plans, les procédures opératoires standards, etc.
5. Collecte et acquisition de données : Cette fonction collecte et met à jour les informations de production utilisées pour le suivi des produits, les historiques de production, etc.
6. Gestion du personnel : Cette fonction permet un suivi « à la minute près » du statut du personnel (présence, production, préparation de matériel, etc.)
7. Gestion de la Qualité : Cette fonction permet une analyse des produits et des processus de production pour assurer une qualité suffisante de la production.
8. Gestion du procédé : Cette fonction est totalement décrite dans les descriptions de la Gestion de la Qualité (7) et du Cheminement des produits et des lots (3). Cette distinction n'est réalisée que pour préciser qu'un système différent peut être utilisé pour réaliser cette fonction.
9. Gestion de la maintenance : Cette fonction classe, dirige, assure la périodicité et gère la disponibilité des outils nécessaires aux opérations de maintenance.
10. Traçabilité produit et généalogie : Cette fonction permet une visibilité sur la disposition spatiale et temporelle de chaque opération.
11. Analyse des performances : Cette fonction permet un suivi « à la minute près » des résultats courants des opérations de production (utilisation des ressources, disponibilité des ressources, temps de production unitaire, conformité à l'ordonnancement prévu, etc.) et une comparaison à l'historique et aux objectifs de performance.

Le niveau encore en dessous (la commande, niveau 1) comprend les applications et matériels nécessaires au contrôle/commande du système automatisé de production. Enfin, le niveau le plus bas (l'opération, niveau 0) contient tous les matériels nécessaires à la détection et à l'actuation : capteurs, actionneurs, etc.

En réalité, une telle décomposition est relativement artificielle. Les frontières entre ces différents niveaux sont souvent mal définies. Ainsi, on voit par exemple se développer aujourd'hui le concept de capteur intelligent, qui place ce dernier entre le et niveau 0 et 1, les ERP (dernier niveau) intègrent de plus en plus de fonctions décrites comme appartenant au MES (avant-dernier niveau), etc.

Aujourd'hui une entreprise ne se pilote plus seulement par les flux informationnels concentrés dans un SI monolithique (comme avec le concept CIM) [Auzelle 09], mais plutôt par l'objet à forte Valeur Ajoutée dont l'entreprise est créatrice : le produit ou le service. Ces changements engendrent des problèmes d'assemblage pour former un SIE cohérent au regard des performances globales à atteindre [Auzelle 08]. En effet, ces SI étant pour la plupart considérés comme des composants « boîte noire », leur configuration doit prendre en compte les propriétés et les fonctionnalités d'un assemblage plus global auquel ils vont participer tout

en conservant les conditions opérationnelles propres à chacun. Cette relation d'interopérabilité, souvent éphémère, amène au développement de **SI** distribués, hétérogènes et autonomes avec comme point commun la partie du **SI** relative au produit.

Le paradigme **SCP** (Figure 2.7) est une alternative pragmatique pour permettre aux entreprises de migrer d'une organisation intégrée vers une organisation adaptable puisque l'originalité de cette approche est de combiner des décisions centralisées prises a priori sur des horizons à moyen, voire à long terme, avec des décisions décentralisées prises « en exécution » sur des horizons à court terme, voire sur événements.

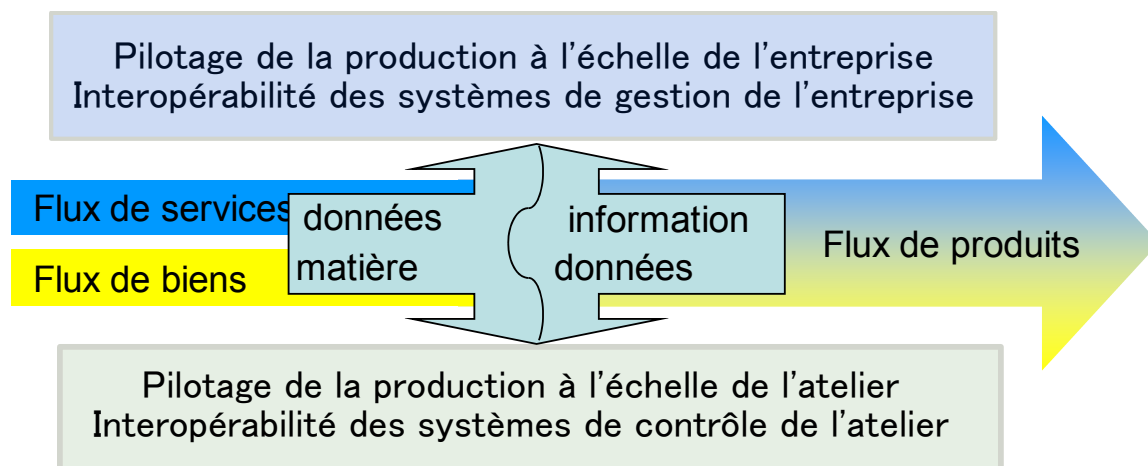


FIGURE 2.7 – Architecture de décision centrée sur le produit inspirée par [Morel 07, Pannequin 07, Auzelle 09]

2.4.2 La notion de Système de Systèmes et le Système d'Information Hospitalier

Le *Système d'Information Hospitalier* (**SIH**) a beaucoup de ressemblances avec un *Système d'Information* (**SI**) d'une entreprise [Huet 09c]. En effet, le logiciel de gestion de la pharmacie peut être comparé à un *Entreprise Resource Planning - Progiciel de gestion intégré* (**ERP**), c'est-à-dire à un progiciel qui permet de gérer l'ensemble des processus d'une entreprise en intégrant l'ensemble de ses fonctions, dont la gestion des ressources humaines, la gestion comptable et financière, l'aide à la décision, mais aussi la vente, la distribution, l'approvisionnement et le commerce électronique.

Kuras [Kuras 06] voit un ensemble de **SI** collaborant entre eux de la façon suivante : « La coopération entre les **SI** internes à chaque entreprise ou entre des **SI** au sein d'un réseau d'entreprises se fait à différents niveaux d'échange, introduisant un facteur d'échelle. » Ce facteur d'échelle est aussi présent dans un **SIH**. Le **SI** d'une entreprise peut être vu comme un *Système de Systèmes* (**SdS**) s'il répond aux critères suivants : autonomie, diversité, appartenance et comportement émergent [Boardman 06]. Ce concept peut s'appliquer aussi pour l'interopérabilité dans un **SIH** où plusieurs logiciels autonomes remplissant des missions différentes doivent collaborer. Par les différentes définitions qui existent d'un système et d'un **SdS** nous proposons les définitions de [Le Moigne 77] et de [Luzeaux 08] (définition 2.6 et 2.7).

Définition 2.6 *Un système est un objet qui, dans un environnement, doté de finalités, exerce une activité et voit sa structure interne évoluer au fil du temps, sans qu'il perde pourtant son identité unique [Le Moigne 77].*

Définition 2.7 *Un système de Systèmes est un assemblage de systèmes pouvant potentiellement être acquis et/ou utilisés indépendamment, pour lequel le concepteur, l'acquéreur et/ou l'utilisateur cherche à maximiser la performance de la chaîne de valeur globale, à un instant donné et pour un ensemble d'assemblages envisageables [Luzeaux 08].*

En se rapprochant de ce concept et en mettant le produit au cœur de ce SdS, [Auzelle 08] nous présente les connexions qui existent entre le produit et les différents composants de ce système (figure 2.8). Il introduit alors la notion d'interopérabilité contrôlée par le produit. C'est dans ce contexte que nous nous inscrivons en comparant le produit à une préparation de la prescription pour un patient donné sur le *Circuit Du Médicament* (CDM). Le CDM a un SI qui lui est propre à l'intérieur du SIH : le *Système d'Information Pharmaceutique* (SIP). Par conséquent, cette préparation sera composée d'une partie physique et d'une partie informationnelle interopérante avec le SIP et les ressources utilisées. Dans l'exemple de la figure 2.8, le produit est connecté à des systèmes d'une entreprise (ES) dont par exemple un ERP, un MES et un SCM. Le produit est donc le pivot de la communication entre les différents niveaux de planification de l'entreprise. Si nousinstancions la vue générale de la figure 2.8 au CDM, nous obtenons la figure 2.9.

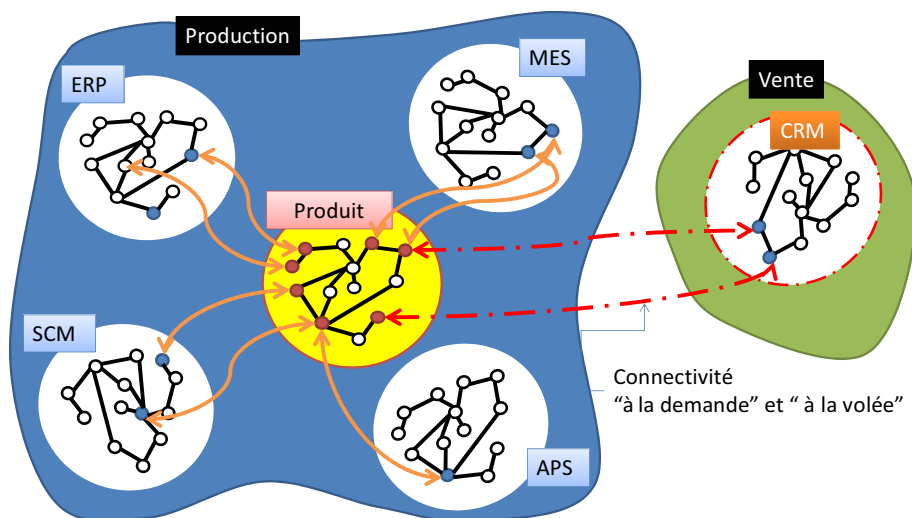


FIGURE 2.8 – Interopérabilité dirigée par le produit [Auzelle 08]

En effet, le produit (la préparation de la prescription) interopère avec le SI du robot et avec le logiciel de gestion de la pharmacie (PHARMA à Clermont-Ferrand). Cette interopérabilité permet au produit une meilleure traçabilité et permet de faire des vérifications plus facilement. Elle ne peut exister que si nous utilisons de nouvelles technologies comme la RFID.

2.4.3 Les modèles d'interopérabilité

Il existe deux approches majeures pour la classification de l'interopérabilité dans les domaines des applications et des systèmes d'informations. La première est une approche basée sur l'étude de l'interconnexion technologique des applications et systèmes interopérant : LISI [AWG 97]. La deuxième se focalise sur l'aspect conceptuel de l'interopérabilité à travers la définition des données et des interfaces d'échange entre les systèmes communiquant : LCIM [Tolk 03].

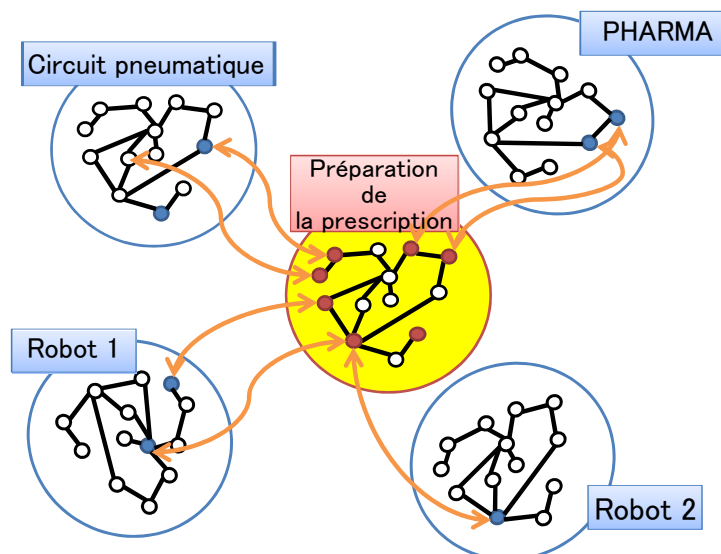


FIGURE 2.9 – Concept d'interopérabilité appliqué au SIP et au CDM

2.4.3.1 Le modèle LISI

Le modèle *Levels Information System Interoperability* (LISI) [AWG 97] est probablement le modèle d'interopérabilité le plus répandu dans le domaine de l'interopérabilité technologique des SI, et même si son développement date de 1998, il reste le modèle de référence dans le domaine. Le modèle LISI aide à développer un profil d'interopérabilité entre systèmes indépendants; la comparaison de ces profils permet d'étudier le degré d'interopérabilité potentielle entre les systèmes. Le modèle LISI définit un modèle de référence composé de cinq niveaux de maturité pour l'interopérabilité, ainsi qu'un ensemble de caractéristiques relatives à ces cinq niveaux.

- 0 **Systèmes isolés** : aucune connexion n'existe entre les applications. Les échanges de données sont effectués manuellement,
- 1 **Systèmes connectés** : les applications sont connectées électroniquement par des liens de type peer-to-peer, leurs fonctions sont complètement séparées, chacune possède son propre environnement et ses propres données,
- 2 **Systèmes distribués** : les applications ont quelques fonctions communes, mais sont totalement indépendantes, leurs données aussi sont séparées. Les échanges de données hétérogènes sont possibles,
- 3 **Systèmes intégrés** : les applications sont séparées mais partagent des données communes. Elles peuvent collaborer de manière sophistiquée,
- 4 **Systèmes universels** : les applications et leurs données sont partagées au sein d'une même entreprise ou entre plusieurs entreprises. Les mécanismes de collaboration entre applications sont très développés.

Les cinq niveaux d'interopérabilité du LISI sont basés sur quatre caractéristiques interdépendantes pour la catégorisation de l'interopérabilité. Ces quatre aspects sont connus sous l'anagramme PAID :

- **P** pour procédures, reflète les procédures, approches, règles, standards et méthodes mis en œuvre pour établir l'échange d'informations entre systèmes,

- **A** pour applications, décrit les applications permettant l'échange, le traitement et la manipulation des données partagées entre différents systèmes,
- **I** pour infrastructures, décrit l'environnement (matériel, réseaux etc.) qui entre en jeu pour assurer l'interaction des systèmes,
- **D** pour données, décrit les formats et protocoles qui permettent l'échange de données, ainsi que l'aspect sémantique permettant l'échange d'information.

La matrice résultant du croisement des cinq niveaux d'interopérabilité et les quatre caractéristiques PAID donne lieu au modèle de référence **LISI** (table 2.1).

Description	Niveau correspondant	Niveau	P	A	I	D
Entreprise	Universel	4	Niveau entreprise	Interactive	Multiples topologies	Modèle d'entreprise
Domaine	Intégré	3	Niveau Domaine	Groupware	Réseaux mondiaux	Modèle du domaine
Fonctionnel	Distribué	2	Niveau application	Desktop Automation	Réseaux locaux	Modèle de programme
Connecté	Peer to Peer	1	Niveau local	Pilote standard de système	Simple connexion	Local
Isolé	Manuel	0	Contrôle d'accès	N/A	Indépendant	Privé

TABLE 2.1 – Modèle de référence de **LISI** [AWG 97]

Le modèle **LISI** offre les bases pour la comparaison et la discussion de l'interopérabilité. Ce cadre est particulièrement intéressant lorsqu'il s'agit de qualifier le niveau de maturité d'un SI, mais il ne fournit pas de guide ou de méthode permettant de le spécifier à des niveaux d'échelles différents et sur des niveaux d'observation différents à travers le temps afin d'en capitaliser ses modèles.

2.4.3.2 Le modèle LCIM

Le modèle *Levels of Conceptual Interoperability Model* (**LCIM**) part du constat qu'il faut de bons modèles composables à un niveau conceptuel pour permettre une interopérabilité significative. Il propose donc de compléter les méthodes de classification technique en se basant sur l'étude conceptuelle de la qualité et de la documentation des interfaces relatives aux données échangées entre des composants interopérants. Le modèle **LCIM** définit cinq niveaux d'interopérabilité [Tolk 03], dont le niveau 0 exprime l'absence totale d'interopérabilité entre les systèmes (figure 2.10).

- **Niveau 0** - Données spécifiques au système : Pas d'interopérabilité entre les deux systèmes. Les données sont utilisées à l'intérieur du système d'une façon propriétaire et sans visibilité externe. Le composant est dit « boîte noire ».

- **Niveau 1** - Données documentées : les données sont documentées en utilisant un protocole commun offrant une interface d'accès. Le composant est dit « boîte noire avec une interface ».
- **Niveau 2** - Données statiques alignées : les données sont documentées en utilisant un modèle de référence commun basé sur une ontologie commune, i.e., le sens des données est décrit d'une manière non ambiguë. Ceci est possible également grâce à l'utilisation de standard de définition de meta-données. Le composant est dit « boîte noire avec une interface standard ».
- **Niveau 3** - Données dynamiques alignées : les données utilisées à l'intérieur même du système (ou composant) sont bien définies, et ce en utilisant des méthodes standards telles que **UML**. Le système est dit « boîte noire derrière une interface » ou encore « boîte blanche ».
- **Niveau 4** - Donnée harmonisée : les connections sémantiques entre les données qui ne sont pas reliées lors de l'exécution du code sont évidentes par la documentation du modèle conceptuel sous-jacent du composant. Ce n'est pas seulement une boîte blanche, car toutes les relations sont définies dans le modèle conceptuel lors d'une implémentation.

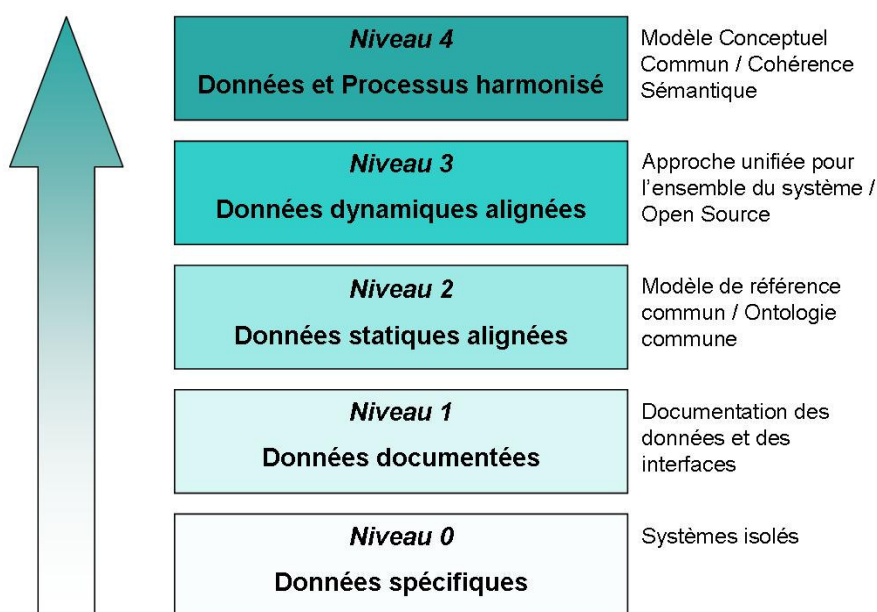


FIGURE 2.10 – Classification de l'interopérabilité orientée données [Tolk 03]

2.5 CONCLUSION

Ce chapitre a été consacré à la description d'un *Système Contrôlé par le Produit* (**SCP**) et de ses caractéristiques. Nous avons commencé par présenter les différentes approches de pilotage : hiérarchique, hétérarchique et hybride. Nous avons alors constaté les limites et les avantages des approches hiérarchiques et hétérarchiques ; nous avons souhaité conserver les avantages et supprimer les inconvénients de ces approches grâce à une approche hybride. Pour ce faire, nous avons proposé d'utiliser le concept d'holon. Nous avons présenté les différentes architectures holoniques et comparé un *Holon Manufacturing System* (**HMS**) par

rapport à un *Système Multi-Agent* (SMA). Ensuite, nous avons exprimé les besoins d'interopérabilité dans une entreprise et dans le milieu hospitalier. Nous avons montré les différents niveaux d'interopérabilité qui peuvent exister dans une entreprise et la problématique de coordination entre SI qui en découle. Pour résoudre cette problématique, la notion de *Système de Systèmes* (SdS) est exposée ainsi que sa correspondance avec le milieu hospitalier. Nous avons mis alors le produit comme pivot entre les différents niveaux d'interopérabilité. Nous avons rejoint alors le concept de SCP avec le paradigme d'interopérabilité contrôlé par le produit. Enfin, nous avons expliqué les modèles d'interopérabilité qui permettent de classifier l'interopérabilité dans une entreprise : le modèle *Levels Information System Interoperability* (LISI) qui la classifie par rapport au degré de connexion et le modèle LCIM qui la classifie suivant la qualité de la documentation.

Un système correspond à une organisation constituée d'un système de pilotage et d'un système opérant [Le Moigne 77]. Le système de pilotage assure la régulation, le système opérant la transformation. Pour des systèmes plus complexes, le système de pilotage est lui-même constitué d'un système informationnel et d'un système décisionnel. Nous avons vu que l'approche de pilotage la plus adaptée était une approche hybride via le concept holonique. Un holon est une entité composée d'un système opérant et d'un système de pilotage, il est donc lui-même un système. Un HMS est donc un SdS. Nous voyons alors un HMS comme un système constitué de holons interagissant entre eux, le HMS interagissant lui-même avec les logiciels des différents niveaux de décision du *Système d'Information d'Entreprise* (SIE). Un produit équipé de technologie d'identification est considéré comme un holon. Ainsi, le paradigme SCP postule que le « Produit », de par ses propriétés qui le rendent « Actif », collabore avec les applications constituant le SIE aux moments opportuns et à la volée pour que ces dernières puissent permettre son évolution durant son cycle de vie et en particulier dans sa phase de production. Si nous nous plaçons dans le contexte hospitalier, nous voyons le *Système d'Information Hospitalier* (SIH) comme un SIE particulier. Ce SIH est composé d'un *Système d'Information Pharmaceutique* (SIP) qui contient toutes les applications utilisées pour le *Circuit Du Médicament* (CDM). Sur le CDM, le produit est considéré comme étant l'ensemble des doses préparées pour un patient ; cet ensemble sera équipé d'une puce RFID. Il va collaborer avec les différentes applications constituant le SIP, exactement comme le produit d'un système de production avec le SIE.

En ce sens, le produit se place au centre du système de production [Gouyon 04]. Il peut alors être considéré comme le pivot devant assurer la cohérence entre les niveaux de planification ERP et d'exécution de la fabrication MES.

Maintenant que nous avons identifié l'organisation des SCP en mettant en exergue ses caractéristiques et ses besoins, nous allons voir dans les chapitres suivants les différents outils de modélisation que l'on rencontre dans la littérature dans le but de pouvoir concevoir un tel système.

3

ÉTUDE DES OUTILS DE MODÉLISATION RELATIVES AUX SYSTÈMES DE PRODUCTION

RÉSUMÉ DU CHAPITRE 3

Dans la littérature, de nombreux auteurs ont proposé des solutions pour modéliser un système de production. Nous proposons d'étudier les plus utilisées dans ce chapitre.

Ce chapitre est donc un état de l'art sur les différents outils existants pour modéliser et concevoir un système de production de biens et de services. Nous élargissons aussi notre vue aux systèmes multi-agent. En effet, nous avons montré que les concepts d'holon et d'agent ont des points communs. Nous commençons ce chapitre en expliquant les différentes approches de modélisation existantes. Nous présentons à la suite :

- les architectures de référence,*
- les méthodologies de modélisation,*
- les méthodes de modélisation,*
- les langages de modélisation.*

Chaque méthodologie, méthode et langage utilisent différentes approches qui sont exposées dans le début de ce chapitre.

Après, nous expliquons comment nous allons faire notre choix entre tous ces outils de modélisation. La dernière section est consacrée à la présentation d'un bilan et le lien avec les chapitres précédents et la suite de la thèse.

SOMMAIRE

3.1	INTRODUCTION	39
3.2	LES ARCHITECTURES DE RÉFÉRENCE	40
3.2.1	Le cadre de Zachman	40
3.2.2	CIMOSA	41
3.2.3	ARIS	42
3.2.4	L'ingénierie dirigée par les modèles (IDM) et l'architecture dirigée par les modèles (MDA)	43
3.3	LES MÉTHODOLOGIES DE MODÉLISATION	44
3.3.1	PERA	45
3.3.2	La méthodologie ASCI	45
3.3.3	Méthodologie Bidirectionnelle de Conception des Systèmes Automatisés (MBCSA)	47
3.3.4	Les méthodologies de conception des Systèmes Multi-Agents	48
3.3.4.1	MaSE	48
3.3.4.2	TROPOS	49
3.3.4.3	GAIA	50
3.4	LES MÉTHODES DE MODÉLISATION	51
3.4.1	IDEF-SADT	51
3.4.2	GRAI-GIM	53
3.4.3	Merise	53
3.4.4	L'environnement MEGA	53
3.5	LES LANGAGES DE MODÉLISATION	54
3.5.1	IEC 61499 - Les Blocs Fonctions	55
3.5.2	Le langage UML et ses extensions	56
3.5.2.1	AUML	56
3.5.2.2	UEML	56
3.5.2.3	SysML	57
3.5.2.4	UML pour le Temps Réel	57
3.5.3	LOTOS	57
3.5.4	L'analyse de déroulement, la norme JIS Z 8206	58
3.5.5	Formalisme pour l'analyse d'un flux de production	58
3.5.6	Les réseaux de Petri	61
3.6	UTILISATION DES MODÈLES	61
3.7	CONCLUSION	61

3.1 INTRODUCTION

Nous nous intéressons dans ce chapitre aux méthodologies, méthodes et langages de modélisation et d'ingénierie. Nous avons vu précédemment la problématique et les concepts utilisés pour la modélisation d'un HMS, nous cherchons alors les outils qui pourraient nous permettre de concevoir et de réorganiser un tel système. Nous nous sommes intéressés aux systèmes de production ainsi qu'au *Système Multi-Agent (SMA)* dont le concept d'holon se rapproche (voir chapitre précédent). Avant tout, nous avons voulu donner une définition de la modélisation (définition 3.1).

Définition 3.1 *La modélisation est une action d'élaboration et de construction intentionnelle, par composition de symboles, de modèles susceptibles de rendre intelligible un phénomène perçu complexe, et d'amplifier le raisonnement de l'acteur projetant une intervention délibérée au sein du phénomène ; raisonnement visant notamment à anticiper les conséquences de ces projets d'action possibles [Le Moigne 77].*

Le résultat d'une modélisation est une représentation de la réalité appelée un modèle (définition 3.2).

Définition 3.2 *Un modèle est un élément conceptuel, visant à la compréhension et au diagnostic, c'est une vue de l'esprit, analytique ou algorithmique représentant des phénomènes et leurs relations.*

Il existe différentes approches pour modéliser un système complexe. Les approches systémiques se focalisent sur l'interaction des systèmes et plus particulièrement sur l'analyse des flux. Chaque système se décompose en sous-systèmes [Le Moigne 77]. Au sein de chaque sous-système, on distingue deux parties : le système de pilotage comprenant le système de décision (chargé de décider des actions à entreprendre) et le système d'information ou logique (qui collecte, mémorise, traite et distribue l'information) ; le système opérant ou physique qui assure les fonctions (fabriquer, livrer, facturer, approvisionner dans le cas d'une industrie). L'approche objet s'est développée dans les années quatre-vingt-dix, d'abord dans le domaine de la programmation, puis dans celui de la conception et de l'analyse des systèmes. La volonté était de modéliser un système tout à la fois d'un point de vue structural, fonctionnel et dynamique, en partant d'un travail d'abstraction. Ces approches doivent être formalisées par un langage permettant de représenter des concepts (vocabulaire) et des règles (grammaire).

Le développement de l'approche objet a créé un intérêt pour l'approche par composant à cause de la nécessité de réutiliser le code (réduction des coûts et du temps). Beaucoup d'auteurs proposent des méthodes pour la conception de composants logiciels et aussi des méthodes de traitement pour assembler, modifier et choisir les composants [Brown 96]. [Blobel 08] utilise cette approche pour un système d'information médical et explique comment créer un « middleware » pour connecter les composants entre eux en correspondance avec la régulation médicale (Health). L'approche par composant comporte deux étapes : une vue en boîte noire des composants qui montre les interactions entre eux et une vue en boîte blanche qui décompose les composants en montrant leurs comportements internes et les éléments structurels [Kisti 05]. [Azevedo 00] a conçu des objets distribués orientés technologie et une architecture basée sur les composants, dans le but d'avoir le plus de flexibilité et de

modularité pour organiser la production et la planification des opérations d'une entreprise distribuée (une compagnie multi-site). Ces deux articles proposent de commencer l'étude par un diagramme de cas d'utilisation (il décrit le système avec un point de vue utilisateur) dans le but d'isoler les sous-systèmes et les composants.

Ces différentes approches sont utilisées par différentes méthodologies, différentes méthodes et différents langages pour pouvoir modéliser un système de production. Dans ce chapitre, nous exposerons en premier lieu les architectures de référence qui permettent de poser les points de repère, puis les méthodologies utilisées pour la conception des systèmes de production de biens, de services et des SMA. Nous verrons ensuite les méthodes et les langages les plus utilisés. Nous expliquerons d'abord les raisons pour lesquelles nous souhaitons utiliser des modèles pour représenter des systèmes de production et enfin nous ferons un bilan et le lien avec les chapitres précédents et futurs.

3.2 LES ARCHITECTURES DE RÉFÉRENCE

La modélisation d'entreprise offre une multitude de méthodes et d'outils développés au cours des dernières années pour répondre aux nombreux besoins de l'industrie tels que l'urbanisation d'un *Système d'Information* (SI), la mise aux normes qualité ou bien encore l'amélioration de la productivité ou des performances ; la partie qui suit ne se veut nullement exhaustive, elle passe en revue les architectures de référence les plus usitées. Selon [Vernadat 02] une architecture de référence fournit un cadre général et des points de repère indiquant aux utilisateurs quels aspects de l'entreprise doivent être pris en compte au cours de la modélisation. Une architecture de référence est un support à une méthode. Suivant la problématique abordée, l'architecture de référence sera considérée comme un canevas (une boîte à outils) qu'il faudra compléter par l'usage de langages de modélisation. Ainsi, il est avantageux de baser une modélisation sur un cadre d'architecture afin de circonscrire la complexité du système à modéliser.

3.2.1 Le cadre de Zachman

Zachman [Sowa 92] propose un cadre de modélisation structuré dont le but est de rendre le plus exhaustif possible l'étude des points de vue des modélisateurs sur le système à étudier. Le cadre de modélisation de Zachman est une matrice à deux dimensions avec 6 colonnes et 6 lignes (table 3.1). Chaque cellule représente une vue partielle d'un système complexe ou non. Les lignes définissent le niveau de modélisation et donc le point de vue du modélisateur : contextuel (vue du planificateur), conceptuel (vue du propriétaire), logique (vue du concepteur), physique (vue du réalisateur), formel (vue des sous-traitants), d'exécution. Les colonnes représentent le point de vue de l'observation du modélisateur : quoi, comment, où, qui, quand, pourquoi. L'union des cellules doit couvrir l'ensemble des points de vue d'un système. Les grands principes qui guident l'application du cadre Zachman comprennent :

- Un système « complet » pouvant être modélisé en répondant à des questions simples (perspectives) comme : pourquoi, qui, quoi, comment, où et quand ?
- Les six perspectives (pourquoi, qui, quoi, comment, où, quand) capturent tous les modèles critiques exigés pour l'analyse, la spécification et le développement d'un système,
- Les contraintes pour chaque perspective sont cumulatives. Celles d'une rangée inférieure sont ajoutées à celles des lignes supérieures pour fournir un nombre croissant de restrictions,

	Data (what)	Function (how)	Network (where)	People (who)	Time (when)	Motivation (why)
Scope (contextual)	List of things important to the enterprise	List of processes the business performs	List of locations where the enterprise operates	List of organizations/ Important to the Business	List of events / Significant to the business	List of business goals / strategies
Enterprise Model (conceptual)	Semantic Model	Business process model	Logistics network	Work Flow	Model Master schedule	Business plan
System Model (logical)	Logical Data model	Application architecture	Distributed system architecture	Human interface architecture	Processing Structure	Business rule model
Technology model (physical)	Physical Data model	System design	System architecture	Presentation architecture	Control structure	Rule design
Detailed representations (out of context)	Data definition	Program	Network architecture	Security architecture	Timing definition	Rule specification
Functioning Enterprise	Data	Function	Network	Organization	Schedule	Strategy

TABLE 3.1 – *Le cadre de Zachman*

- Les colonnes représentent différentes abstractions dans un effort de réduire la complexité de n'importe quel modèle simple qui est établi,
- Les colonnes n'ont pas d'ordre,
- Les modèles dans chaque colonne sont uniques,
- Chaque ligne représente une perspective unique,
- Chaque cellule est unique en son genre,
- La logique est récursive.

Un des avantages majeur du cadre de modélisation de Zachman est d'offrir au modélisateur une vision complète du système complexe qu'il doit modéliser. Ainsi, ce cadre fournit un guide (ou filtre) avec des croisements de points de vue simples à interpréter (la matrice 6X6) et permet d'avoir une vision complète rapidement accessible des objets de modélisation d'un système à faire. Le cadre de Zachman n'est qu'une structure pour décrire une entreprise, il nécessite donc une méthodologie (un process d'utilisation) pour pouvoir être utilisé pour la conception.

3.2.2 CIMOSA

CIM Open System Architecture (**CIMOSA**) [AMICE 93, Kosanke 99], est une architecture pour construire des systèmes intégrés de production. Elle a été développée par le Consortium AMICE dans le cadre des projets ESPRIT.

Le cadre de modélisation formalise trois principes fondamentaux pour la modélisation en

entreprise suivant une structure à trois axes, communément appelée cube CIMOSA, représentée à la figure 3.1.

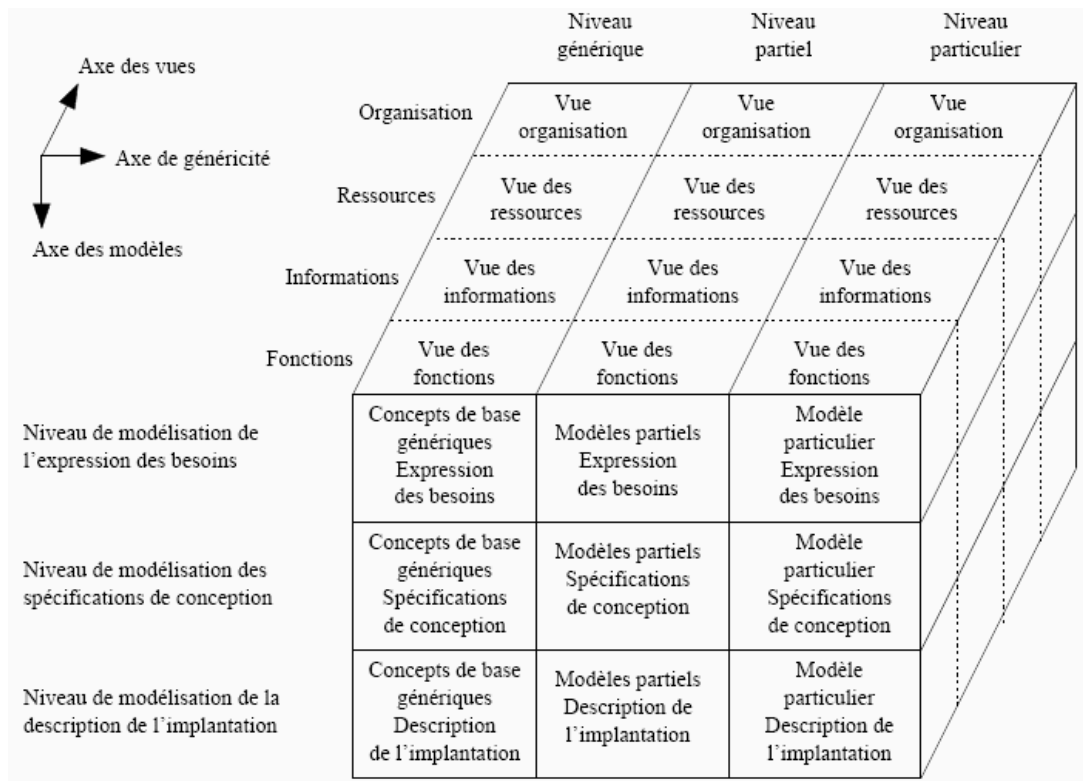


FIGURE 3.1 – Le cadre de modélisation de CIMOSA

Elle est un des seuls cadres de modélisation qui, avec la matrice de « Zachman », permet de passer en revue tous les aspects de spécification d'un projet. Les trois axes retenus sont décomposés comme suit :

- L'axe des vues, qui propose de modéliser d'abord les besoins de l'entreprise,
- L'axe des modèles, qui invite à modéliser d'abord les besoins de l'entreprise, puis les spécifications de conception et enfin la description de l'implantation,
- L'axe de la généricité, qui suggère de construire le modèle particulier de l'entreprise à partir de modèles partiels, eux-mêmes exprimés en termes de construction générique de base.

Malheureusement, l'absence de guide pour le formalisme de la description de chaque cellule et le grand nombre de cellules de la matrice rendent difficile sa mise en œuvre qui peut vite devenir inexploitable.

3.2.3 ARIS

La conception de l'ARchitecture de Systèmes d'Information Intégrés (ARIS) repose sur un concept d'intégration dicté par une vision globale des processus de l'entreprise [Scheer 94]. La conception de l'architecture se base tout d'abord sur un modèle développé pour les processus d'entreprise et contenant toutes les caractéristiques principales nécessaires à la description de processus d'entreprise. Le modèle complexe qui en résulte est décomposé en plusieurs vues :

- la vue de fonction,

- la vue d'organisation,
- la vue de données,
- la vue de prestations,
- la vue de contrôle ou la vue de gestion.

Cette décomposition par vues permet alors de procéder à la description du contenu de ces vues à l'aide de méthodes spécialement adaptées à chacune d'elles sans qu'il soit nécessaire de tenir compte des multiples relations et liens que ces vues peuvent entretenir entre elles. Les relations entre les vues sont ensuite prises en compte et regroupées sans redondances pour une vue générale des chaînes de processus.

La deuxième étape permettant de réduire la complexité de l'architecture consiste à distinguer différents niveaux descriptifs. Les diverses méthodes descriptives appliquées aux systèmes d'information sont classées selon un concept Life-Cycle en fonction de leur degré de rapprochement avec les techniques de traitement de l'information. Ceci permet d'obtenir une description approfondie de tous les aspects, depuis la problématique de gestion d'entreprise jusqu'à la transposition technique. Le concept **ARIS** fournit ainsi un cadre dans lequel des systèmes d'information intégrés peuvent être développés et optimisés et dans lequel la transposition de ces systèmes peut être décrite.

C'est en particulier l'accentuation du niveau descriptif spécialisé qui permet au concept **ARIS** de jouer un rôle d'orientation lors de l'élaboration, de l'analyse et de l'évaluation de chaînes de processus économiques.

L'avantage principal de **ARIS** réside dans les attributs caractérisant un processus : des attributs de coûts et des attributs de temps sont donnés à chaque processus et à chaque activité du processus. Les attributs de coûts sont donnés par les frais de personnel, les frais de matières et les autres coûts. Son inconvénient est que **ARIS** est adaptée à l'analyse et la conception de système d'information mais pas du tout adaptée pour les systèmes composés aussi de partie physique.

3.2.4 L'ingénierie dirigée par les modèles (IDM) et l'architecture dirigée par les modèles (MDA)

L'Ingénierie Dirigée par les Modèles (IDM) préconise une génération tardive du code avec plusieurs niveaux d'abstraction, chaque niveau d'abstraction étant représenté par un modèle. Le modèle d'un certain niveau donne les caractéristiques du modèle du niveau inférieur. Un modèle de niveau d'abstraction inférieur doit, par conséquent lui être conforme. Ce modèle de niveau supérieur est souvent appelé « méta-modèle ». La génération tardive du code ne permet de se conformer à une technologie cible qu'au dernier moment et permet la conception de modèles adaptables et réutilisables pour différentes technologies et plateformes. Un autre point clé de l'**IDM** est la transformation de modèles, elle permet de passer d'un modèle source décrit à un certain niveau d'abstraction à un modèle destination décrit éventuellement à un autre niveau d'abstraction. Ces modèles source et destination sont conformes à leur méta-modèle respectif (méta-modèles source et destination) et le passage de l'un à l'autre (i.e. la transformation) est décrit par des règles de transformation. Le *Model Driven Architecture - Architecture dirigée par les modèles (MDA)* est une variante particulière de l'**IDM**. D'autres variantes de l'**IDM** ont été développées, par exemple par Microsoft (DSL Tools).

Dans l'approche **MDA**, l'**OMG** [OMG 03] a défini une typologie de modèles, ainsi qu'un ensemble de relations de transformation qui permettent de passer de l'un à l'autre. Les trois principaux types de modèles définis sont les suivants :

- **CIM (Computation Independant Model)** Aussi appelé modèle de domaine ou modèle

métier, le CIM capture les exigences en termes de besoins et décrit la situation dans laquelle le système sera utilisé. Son but est d'aider à la compréhension du problème mais aussi de fixer un vocabulaire commun pour un domaine particulier. Dans la pratique, l'appellation « CIM » est très peu utilisée.

- **PIM (Platform Independent Model)** Le **PIM** décrit le système indépendamment de la plate-forme cible sur laquelle il s'exécutera. Il présente donc une vue fonctionnelle détaillée du système, sans détails techniques. Il peut être raffiné progressivement jusqu'à intégrer des détails d'architecture spécifiques à un type de plate-forme (machine virtuelle, système d'exploitation, etc.) mais il doit rester technologiquement neutre.
- Une **PSM (Platform Specific Model)** Le **PSM** est le résultat de la combinaison du **PIM** et des caractéristiques d'une plateforme. Il représente une vue technique détaillée du système. Il peut exister avec différents niveaux de détail. Dans sa forme la plus détaillée, il sert de base à la génération de l'implémentation.

La relation entre ces trois types de modèle est montrée sur la figure 3.2.

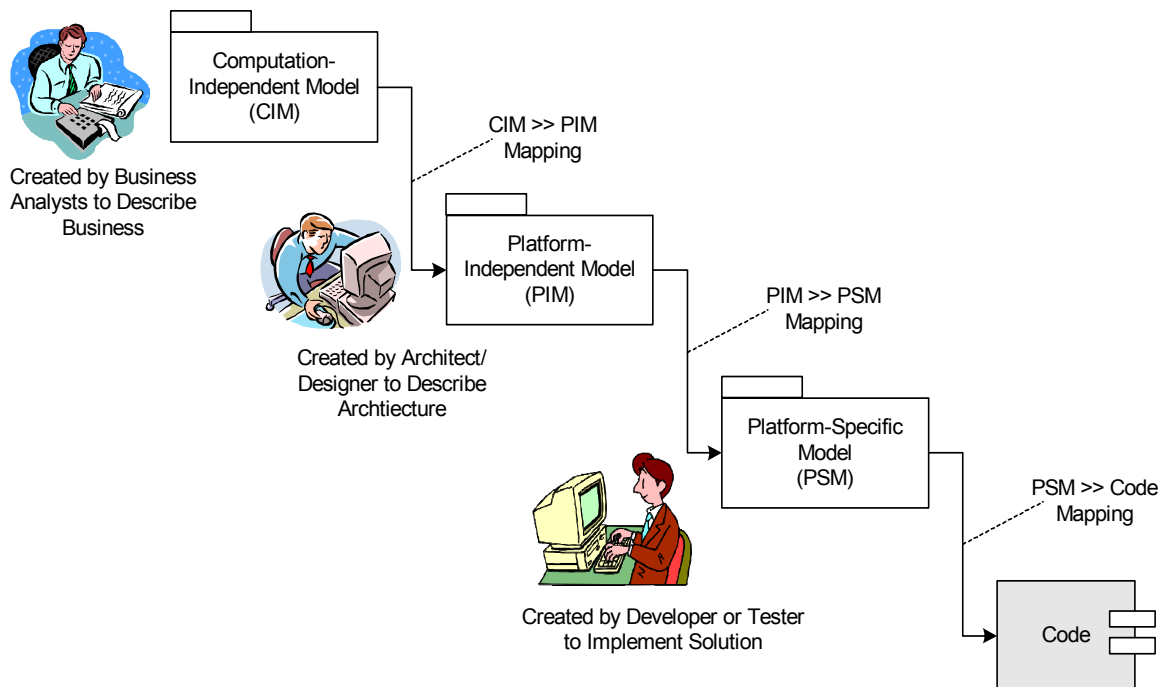


FIGURE 3.2 – Les différents niveaux des modèles de la démarche MDA [Frankel 03]

La démarche MDA propose un « mapping », pour passer d'un PIM à un PSM, c'est-à-dire des règles de transformation pour passer d'un modèle indépendant d'une plateforme à un modèle spécifique d'une plateforme. [Naumenko 03, Baïna 06b] proposent une décomposition en 4 niveaux pour effectuer ces transformations (méta-métamodèle, méta-modèle, modèle, univers du discours).

3.3 LES MÉTHODOLOGIES DE MODÉLISATION

Une méthodologie (définition 3.3) de modélisation permet de donner un guide général à suivre sans donner précisément les outils qui devront être utilisés. Elles peuvent être utilisées avec des cadres d'architectures présentés dans la section précédente.

Définition 3.3 Une méthodologie de modélisation est un ensemble de méthodes, d'outils, d'approches et de concepts permettant de modéliser un système [Chauvet 09].

3.3.1 PERA

Purdue Enterprise Reference Architecture (PERA) [Williams 94], est une méthodologie complète d'ingénierie des environnements industriels développée à l'université de Purdue aux USA. Elle tient compte des aspects humains. Elle peut être généralisée au développement de tout système d'entreprise (système industriel, atelier, usine ou département de toute nature). La méthodologie définit cinq phases du cycle de vie d'une entité industrielle depuis sa conceptualisation jusqu'à sa mise en opération en passant par les phases de conception :

- phase de conceptualisation,
- phase de définition,
- phase de conception,
- phase d'installation et de construction,
- phase opérationnelle et de maintenance.

La phase de conceptualisation peut se diviser en deux étapes :

- l'étape d'identification,
- l'étape de concepts.

Pour résumer, PERA est une méthodologie centrée sur les fonctions mises en œuvre par l'être humain dans le domaine de l'intégration des entreprises.

3.3.2 La méthodologie ASCI

La méthodologie *Analyse, Spécification, Conception, Implémentation* (ASCI) a été développée pour permettre la conception d'outils d'aide à la décision appelés modèles d'action (modèles de simulation, modèles métaheuristiques, modèles analytiques,...) [Gourgand 91, Gourgand 92]. Elle permet de concevoir la modélisation d'une classe de systèmes, le modèle générique de connaissance de cette classe, et de réaliser une bibliothèque de composants logiciels qui est exploitée pour générer un modèle d'action (programme informatique) pour un système de la classe. Ces principales étapes sont décrites sur la figure 3.3.

Le cadre conceptuel de la méthodologie de modélisation ASCI s'appuie sur un processus de modélisation séparant explicitement le recueil de la connaissance de son exploitation. Elle propose une vision systémique qui préconise la décomposition hiérarchique d'un système avec le paradigme objet en trois sous-systèmes communicants et complémentaires (les sous-systèmes logique, physique et décisionnel).

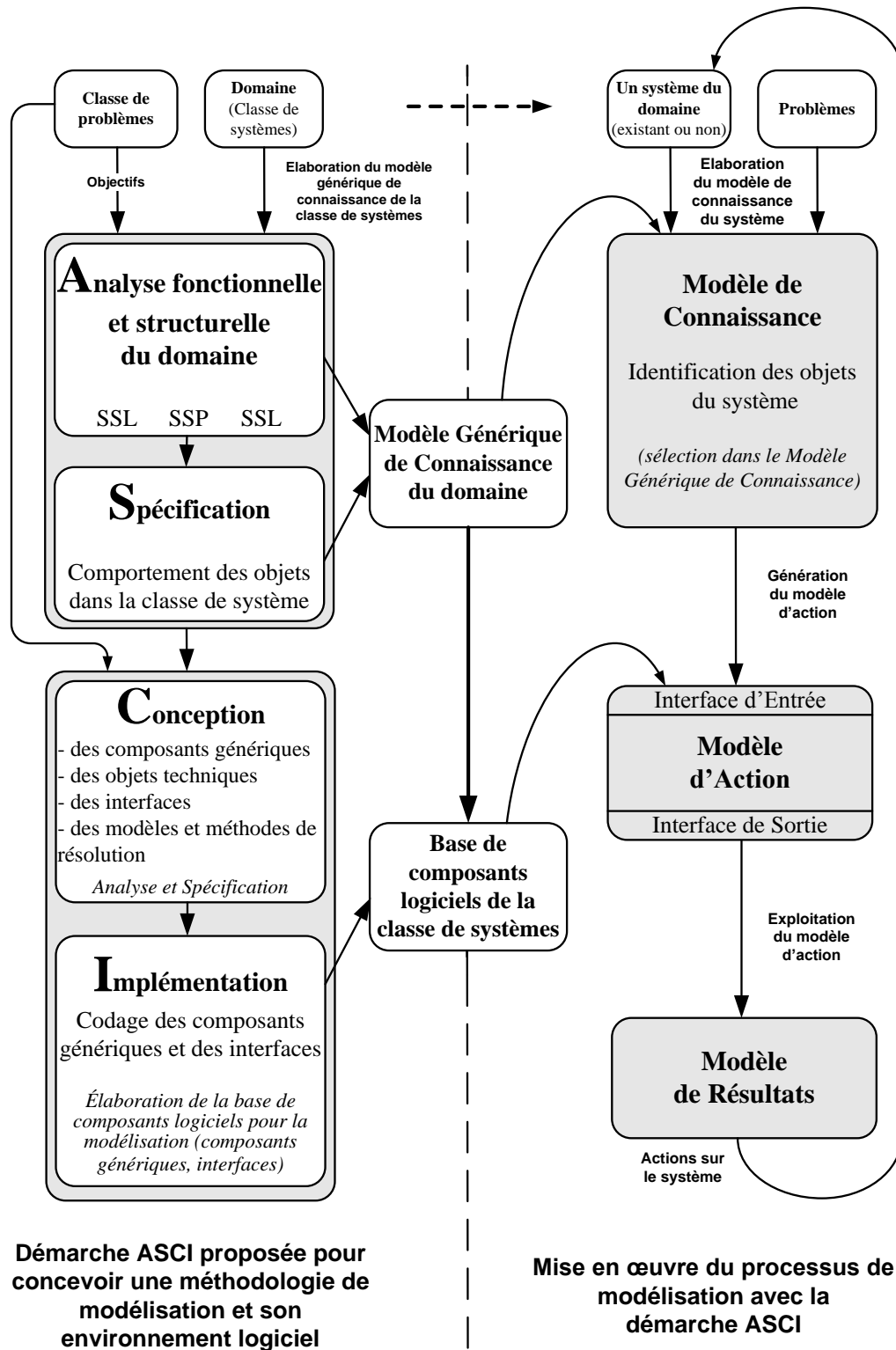


FIGURE 3.3 – La méthodologie ASCI

3.3.3 Méthodologie Bidirectionnelle de Conception des Systèmes Automatisés (MBCSA)

L'approche par composants a été développée pour le génie logiciel pour permettre une réutilisation du code [Brown 96]. Elle est consécutive à l'utilisation de l'Ingénierie Dirigée par les Modèles (IDM) et du Model Driven Architecture - Architecture dirigée par les modèles (MDA) [OMG 03]. Une méthodologie basée sur l'approche composant et le MDA a été développée dernièrement pour la conception des systèmes automatisés : la Méthodologie Bidirectionnelle de Conception des Systèmes Automatisés (MBCSA) [Chiron 09]. Elle s'appuie sur le concept d'objet d'automatisme multi-facettes. Comme dans [Servat 05], la MBCSA s'intéresse à la modélisation structurelle du composant en deux vues principales : une vue en boîte noire décrivant les services fournis et requis par ce composant, découlant de l'attribution des collaborations ; et une vue en boîte blanche qui décrit l'assemblage des composants internes le constituant. La MBCSA est un exemple d'utilisation de la démarche MDA avec l'utilisation d'UML et de SysML. Elle a été développée au sein du LIMOS par Fabien Chiron [Chiron 09]. Il préconise la séparation des spécifications en deux mouvements distincts à savoir :

- Une **Analyse Objet Montante (AOM)** ne se préoccupant pas du contexte de réalisation et se concentrant sur la spécification d'entités élémentaires,
- Une **Analyse Objet Descendante (AOD)** pour la réalisation de projet avec la réutilisation obligatoire d'éléments définis dans le premier cas.

Cette méthodologie est résumée sur la figure 3.4.

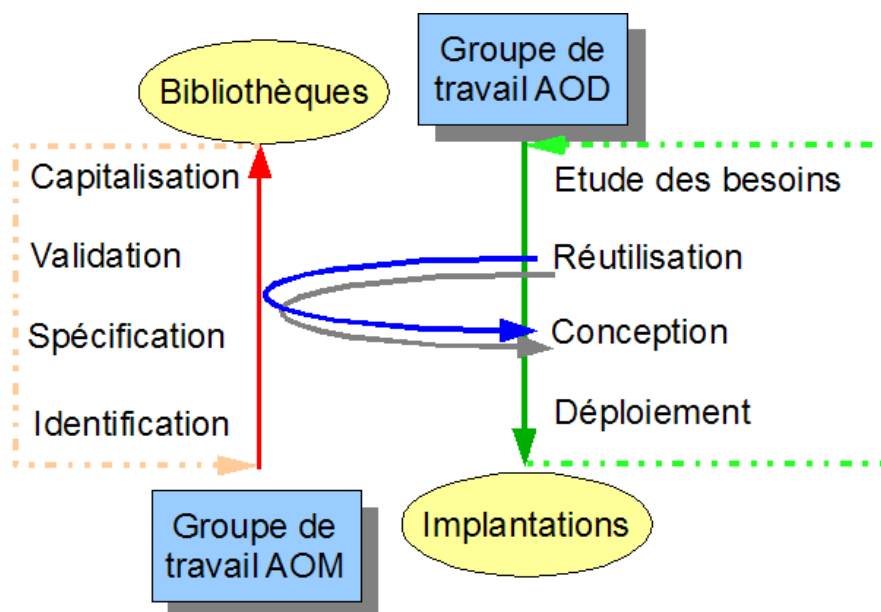


FIGURE 3.4 – Processus d'analyse bidirectionnel [Chiron 09]

Elle montre les différentes étapes pour chacun des deux mouvements. L'AOM passe par les phases d'identification, de spécification détaillée, de validation et de test, puis de capitalisation dans des bibliothèques. Les participants de l'AOD travaillant sur un projet donné, partent d'une analyse des besoins, puis choisissent les composants à réutiliser, les connectent dans une phase de conception, puis les déploient vers les plates-formes cibles.

3.3.4 Les méthodologies de conception des Systèmes Multi-Agents

Comme nous l'avons expliqué dans le chapitre précédent, le concept d'holon et la technologie développée dans les SMA ont certains points communs. Nous nous sommes donc intéressé aussi aux méthodologies les plus répandues pour la conception de SMA afin de vérifier leurs applicabilités dans la conception de systèmes holoniques (les différences entre agent et holon ont été développées dans la section 2.3.3).

3.3.4.1 MaSE

Multiagent Systems Engineering (MaSE) est une approche détaillée d'analyse et de conception des SMA, elle utilise des modèles graphiques dérivés des modèles UML standard pour décrire les types d'agents dans un système et leurs interfaces avec les autres agents [Deloach 01]. MaSE montre les SMA comme une extension du paradigme orienté-objet où les agents sont des objets spécifiés. MaSE est aussi la base de « agentTool », un outil graphique de développement des SMA supportant toutes les étapes de MaSE. Cette méthodologie est composée de deux phases : une phase d'analyse et une phase de spécification, chaque phase consiste en plusieurs étapes, voir (figure 3.5).

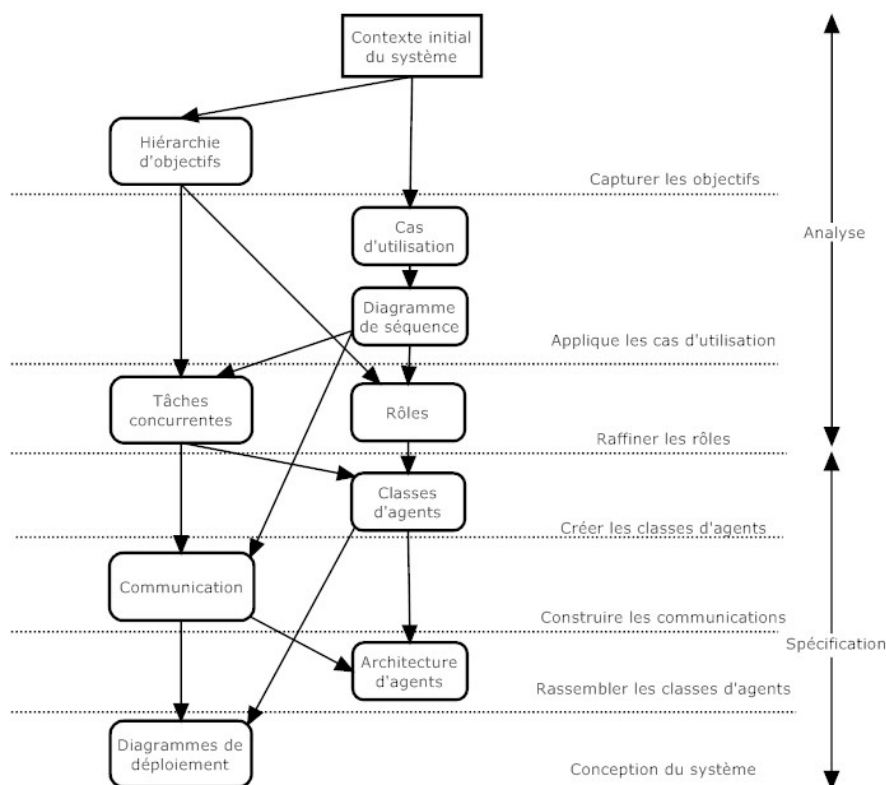


FIGURE 3.5 – Les phases de MaSE [Ali 09]

MaSE sert de base à la méthodologie *Organization-based Multiagent System Engineering* (O-MaSE) [Garcia-Ojeda 08]. O-MaSE est composé de trois structures basiques :

1. un méta-modèle
2. un ensemble de fragments de méthodes

3. un ensemble de directive

L'objectif de **O-MaSE** Process Framework est de guider la conception d'un processus personnalisé orienté agent en utilisant des fragments de méthodes basés sur un méta-modèle commun. Les deux phases de **MaSE** (Analyse et Conception) sont précédées par une phase d'analyse des besoins (Requirements). Elle est basée aussi sur l'OPEN Process Framework qui préconise une approche à 3 niveaux (figure 3.6).

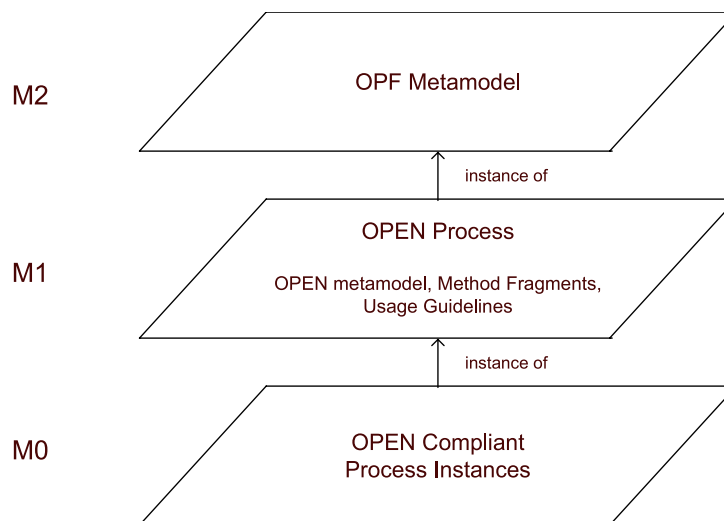


FIGURE 3.6 – OPEN Process Framework [Garcia-Ojeda 08]

Cette méthodologie est parfaitement adaptée à la conception de **SMA** mais demande de nombreuses modifications pour pouvoir être utilisée sur un système holonique (gestion d'une hiérarchie entre holons et d'une partie physique). Par contre, la démarche proposée qui part d'une analyse des objectifs jusqu'à la conception avec des diagrammes orientés objet adaptés à chaque étape nous semble intéressante.

3.3.4.2 TROPOS

Cette méthodologie est composée de cinq phases pour développer un système multi-agents : l'analyse préliminaire des besoins, l'analyse avancée des besoins, la conception architecturale, la conception détaillée et l'implémentation [Bresciani 04].

1. analyse préliminaire des besoins : comprendre le problème en étudiant ses configurations organisationnelles, faire une analyse orientée-but et modéliser les intentions de chaque acteur comme buts. Les acteurs étant des entités sociales dépendantes les unes des autres pour exécuter des tâches, fournir des services et atteindre des objectifs, ces dépendances et relations sont représentées dans un modèle stratégique de dépendance.
2. analyse avancée des besoins : décrire le système dans son environnement opérationnel, construire un modèle stratégique rationnel décrivant comment réaliser les buts identifiés dans l'étape précédente, ce modèle est un graphe contenant quatre types de noeuds : objectif, tâche, ressource et objectif affiné, et deux types de liens : décomposition et méthode.
3. conception architecturale : définir l'architecture globale du système en termes de sous-systèmes inter-connectés via les données, le contrôle et les dépendances.

4. conception détaillée : définir le comportement de chaque composant et les communications entre les agents. Tropos adopte des langages de communication existant comme FIPA-ACL et des diagrammes UML et AUML.
5. Implémentation : fait le lien entre les spécifications de la plateforme d'implémentation et les notions développées dans la phase précédente de conception détaillée.

Les phases d'analyse des besoins sont effectuées avec les diagrammes assez proches des cas d'utilisation d'UML, les phases de conception sont modélisées avec les diagrammes d'activités d'UML et les diagrammes d'interaction d'AUML. L'architecture de TROPOS est définie en quatre niveaux (table 3.2).

Niveau	Description
Méta-méta-modèle	Éléments structurels du langage
Méta-modèle	Une instance du méta-méta-modèle Définit les notions du niveau de connaissance
Domaine	Une instance du méta-modèle Application à une entité d'un domaine
Instance	Instanciations des éléments du domaine

TABLE 3.2 – Les quatre niveaux d'architecture de TROPOS [Bresciani 04]

L'inconvénient est que TROPOS est une méthodologie de développement de logiciel orientée agent. Elle nécessiterait de nombreuses adaptations pour pouvoir être appliquée à un holon (qui a une partie matérielle). Elle part de l'analyse du rôle des acteurs du système comme dans l'approche par composant.

3.3.4.3 GAIA

Gaia [Wooldridge 00] était la première méthodologie complète proposée pour l'analyse et la conception des SMA, la version d'origine de Gaia était plutôt pour les systèmes fermés et elle a adopté des techniques de notation non standard. Les différentes étapes préconisées par cette méthodologie sont données sur la figure 3.7.

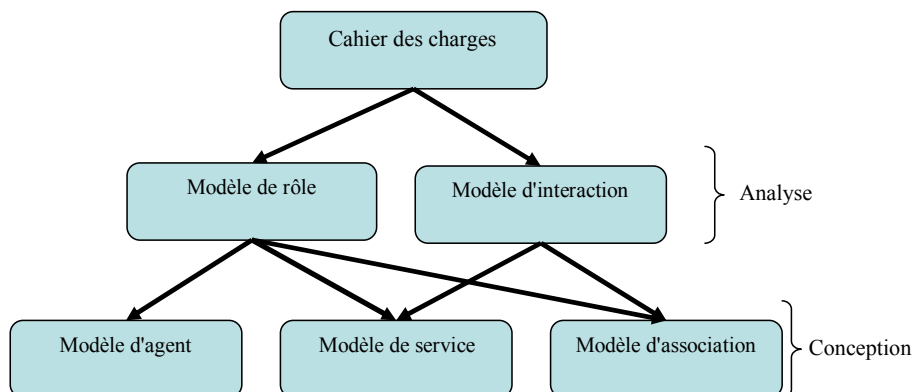


FIGURE 3.7 – La méthodologie Gaia [Ali 09]

La phase d'analyse de Gaia permet d'identifier les rôles du système étudié et de modéliser leurs interactions. Pour chaque rôle elle décrit le comportement de l'agent dans un schéma semi-formel, l'ensemble de ces schémas définit le rôle complet du modèle. Dans cette phase, trois types de modèles sont à produire : le modèle d'agents spécifiant les types d'agents, le modèle de service spécifiant les services à implémenter par les types d'agents et le modèle d'association décrivant les liens de communication entre les agents. Pour construire le modèle d'agent, les rôles seront associés aux agents. Le modèle de service montre les services que chaque type d'agent fournit, ces services sont dérivés à partir des activités et des protocoles des rôles. Les limitations de Gaia sont qu'elle ne peut pas modéliser des aspects sociaux importants, ni des structures organisationnelles d'agents dans le système. En plus Gaia suppose que tous les agents sont coopératifs. Pour ces raisons, Gaia ne convient pas aux systèmes ouverts. Gaia utilise des notations spécifiques pour représenter les rôles et les protocoles. Malgré la simplicité de ces notations, elles ne sont pas suffisantes pour exprimer des problèmes complexes comme les protocoles d'interaction multi-phases. Plusieurs extensions de cette méthodologie ont été proposées pour surmonter les limitations de la version d'origine comme par exemple [Cernuzzi 06] qui propose des solutions pour améliorer l'adaptabilité de cette méthodologie tout au long du cycle de vie du SMA.

3.4 LES MÉTHODES DE MODÉLISATION

Une méthode de modélisation explique comment concevoir et structurer un modèle de connaissance d'un système (une cartographie des processus) à partir d'un formalisme donné et d'une ou plusieurs approches.

Définition 3.4 *Une méthode est une technique de résolution de problème caractérisée par un ensemble de règles bien définies qui conduisent, pour le problème, à une résolution correcte [Calvez 90].*

D'après [Pierreval 90], une méthode doit permettre de :

- guider l'analyste dans sa démarche,
- faciliter la compréhension du système étudié,
- diagnostiquer les anomalies du système étudié,
- communiquer avec les acteurs concernés relevant de différents domaines ainsi que le concepteur avec l'utilisateur.

Nous présentons ainsi SADT/IDEF_x, GRAI, MERISE, MEGA qui sont des méthodes de modélisation.

3.4.1 IDEF-SADT

Le langage *Integrated computer aided manufacturing DEFinition language* (IDEF) a été mis au point par l'US Air Force [Mayer 92] et s'est rapidement popularisé avec notamment son utilisation au sein du Département de la Défense américain (DoD). A partir de ce langage, différentes méthodes ont été construites pour modéliser différents domaines d'application entourant la spécification d'un système : IDEF₀ décomposition des activités, IDEF₁ flux d'informations, IDEF_{1x} structures de données, IDEF₂ simulation, IDEF₃ description des processus, IDEF₄ conception orientée objet, IDEF₅ description d'ontologies, IDEF₆ rationalités conceptuelles, IDEF₇ méthode d'audit pour les systèmes d'information, IDEF₈ interfaces utilisateur, IDEF₉ conception dirigée par les scénarii des systèmes d'information, IDEF₁₀ architectures

d'implantation, IDEF11 artefacts informationnels, IDEF12 organisation du système, IDEF 13 formalisme tri-schémas, IDEF14 conception de réseaux. Comme on peut le voir, selon le domaine d'utilisation, on a, à notre disposition, une multitude de méthodes suivant le domaine dans lequel on se trouve. En industrie, on simplifie souvent cette vision normalement constituée de 13 facettes et on se sert généralement des méthodes IDEF0, IDEF1 et IDEF3.

IDF0 s'est diffusé dans les services méthodes d'un nombre important d'entreprises européennes comme l'une des méthodes les plus efficaces d'analyse descendante, permettant de décrire un système de façon systémique et hiérarchique comme sur la Figure 3.8¹.

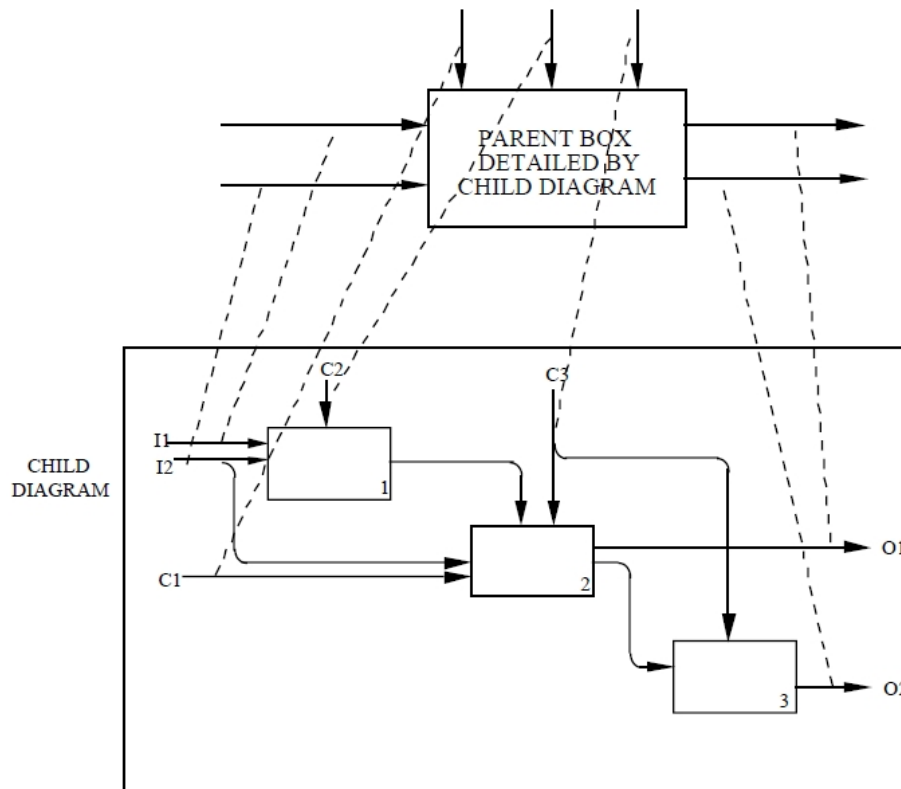


FIGURE 3.8 – Analyse descendante systémique avec IDEF0/SADT

Cette méthode s'inspire fortement du très répandu diagramme SADT (Structured Analysis and Design Technique) mis au point dans la société Softech [Ross 77]. Les briques représentent des fonctions elles-mêmes images d'activités, de processus ou de transformations à l'intérieur du système. Une boîte fonctionnelle appelée Activité est reliée aux autres boîtes et à l'environnement extérieur par l'intermédiaire de flèches puis peut être décomposée à son tour à un niveau inférieur. La popularité de cette représentation vient du fait qu'elle constitue une approche cohérente et non-ambiguë de décomposition d'un système permettant une communication claire entre les analystes, les concepteurs et les utilisateurs du système. On peut remarquer des similitudes avec l'approche par composant comme par exemple le concept de boîte noire pour les processus parents et boîte blanche pour le détail par des diagrammes « enfants ».

1. extrait de www.idf.com (accessible au 8 avril 2010)

3.4.2 GRAI-GIM

La méthode *Graphes à Résultats et Activités Inter-reliés* (**GRAI**) et son extension *GRAI Integrated Methodology* (**GIM**) ont été mises au point à l'université de Bordeaux [Doumeingts 84, Marcotte 94]. La méthode **GRAI** permet la modélisation et l'analyse des systèmes de décision des entreprises de production des biens ou de services. Elle s'appuie sur une description du système en suivant plusieurs approches :

- Le modèle conceptuel de référence, qui décrit le système sous trois angles différents : l'angle gestion de production, l'angle structure du système de production, et l'angle structure décisionnelle du système de production qui aboutit à la définition des centres de décision.
- La grille **GRAI** qui permet de représenter la structure décisionnelle périodique du système de pilotage avec la planification de la production des produits, la gestion de ces produits en terme d'approvisionnement des nomenclatures, la gestion des ressources de production mises en œuvre pour le traitement et le transport des produits (par ressources on entend également les ressources humaines).
- Les réseaux **GRAI** qui reprennent les centres de décision identifiés dans la grille et guident l'utilisateur dans l'identification d'activités, dans le sens de processus de transformation, en les classant en deux catégories : activités de décision et activités d'exécution.

3.4.3 Merise

La *Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise* (**MERISE**) [Tardieu 83] orientée « systèmes d'informations », a rencontré un succès certain dans les années 80. La conception du système d'information repose sur un cycle d'abstraction composé d'étapes afin d'aboutir à un système d'information fonctionnel proche de la réalité. La méthode **MERISE** préconise d'analyser séparément données et traitements dans chaque étape. Le passage à une autre étape correspond à un point de validation et de décision. Il s'agit donc de valider une à une chacune des étapes en prenant en compte les résultats de l'étape précédente.

Cette méthode (en particulier son modèle entité-relation) a été très utilisée avant de voir sa cote décroître au sein des architectes de **SI** au profit d'**UML** notamment dont elle est un des piliers. Une intégration de l'approche objet dans la méthode a également vu le jour avec le projet MERISE+ [Espinasse 94], se proposant de se rapprocher de la méthode orientée objet HOOD (Hierarchical Object Oriented Design). Dans une première étape d'analyse on utilise les deux méthodes (MERISE original et HOOD), ensuite, on compare les spécifications résultantes selon des axes et critères jugés pertinents.

Une telle démarche est trop rigide à cause des nombreux points de validation nécessitant de remonter au système général à la fin de chaque cycle d'abstraction.

3.4.4 L'environnement MEGA

MEGA édité par MEGA International² s'articule autour d'un référentiel qui regroupe trois produits d'analyse et de conception : MEGA Process, MEGA Architecture et MEGA Designer.

- MEGA Process pour cartographier les chaînes de valeur de l'organisation,
- MEGA Architecture pour cartographier et urbaniser les systèmes d'information,

2. <http://www.mega.com> (accessible au 27 octobre 2010)

- MEGA Designer pour réaliser la conception détaillée des systèmes et applications informatiques.

L'outil MEGA permet d'accompagner les managers dans leurs programmes d'Architecture d'Entreprise. Le résultat est un référentiel d'entreprise apportant une vision claire et synthétique des processus métiers et des ressources qu'ils mettent en jeu. Il est construit par l'adoption et l'enrichissement des standards du marché ; ses méta-modèles et principes de modélisation sont publiés et reconnus par les organismes de standardisation internationaux.

Il faut représenter, analyser et corréler ces sujets sans se perdre dans les détails. Cet outil offre un cadre méthodologique ouvert, détaillant chaque étape d'un projet de réalisation d'un référentiel d'entreprise, ainsi que l'ensemble des livrables à produire. La flexibilité de cet outil assure la communication de l'ensemble des documentations aux utilisateurs. Ses fonctionnalités garantissent la cohérence et l'intégrité documentaire indispensables à ce type de démarche.

L'outil regroupe les différentes perspectives de l'architecture d'entreprise dans un référentiel commun en articulant les différents modèles :

- Identification des chaînes de valeur de l'entreprise (les processus métiers),
- Identification de l'architecture applicative par une analyse orientée service,
- Alignement des ressources informatiques sur les processus métiers par les techniques d'urbanisation,
- Alignement de l'organisation sur les processus métiers et sur les ressources informatiques mises à disposition : la conduite du changement,
- Approche centrée sur l'analyse des flux pour une mise en cohérence de la gestion de l'information dans l'entreprise.

Cet outil est centré sur la modélisation des flux d'information. Il ne propose rien pour la modélisation des flux physiques.

3.5 LES LANGAGES DE MODÉLISATION

Les langages de modélisation sont utilisés pour permettre d'analyser et de spécifier le fonctionnement d'un système complexe. Ils sont très utiles pour permettre une communication avec les acteurs de ce type de processus. Ils doivent donc être simples et facilement adaptables. Nous présentons ici les modèles graphiques les plus répandus. Ces langages sont livrés sans guide d'utilisation. Par conséquent, ces langages doivent être utilisés avec des méthodes et des méthodologies, telles celles présentées dans les sections ci-dessus, pour pouvoir être utilisés de façon cohérente. Certains formalismes peuvent aussi représenter les processus que ce soit les flux de production ou les flux de travail. D'autres langages sont adaptés à la modélisation de processus ou de système temps réel. Un processus peut être défini de la manière suivante :

Définition 3.5 *Un processus est un ensemble d'activités qui s'enchaînent de manière chronologique pour atteindre un objectif, généralement délivrer un produit ou un service, dans le contexte d'une organisation de travail.*

3.5.1 IEC 61499 - Les Blocs Fonctions

Les *Function Block* (**FB**) [IEC61499 05] permettent d'encapsuler des logiques complexes au sein d'un programme. Fabien Chiron [Chiron 09] utilisent ces éléments pour représenter le lien entre une approche de modélisation objet à l'aide d'**UML** et la génération d'objets d'automatismes directement déployables sur un *Automate Programmable Industriel* (**API**). En plus de variables d'Entrée/Sortie situées sur leurs interfaces externes de données, les blocs disposent d'un ensemble de variables internes accessibles ou non par le programme parent selon les droits d'accès imposés. On retrouve sur la Figure 3.9 une vue schématique d'un **FB** avec l'encapsulation dans une boîte noire d'un comportement interne programmé dans l'un des langages de la norme IEC 61131.

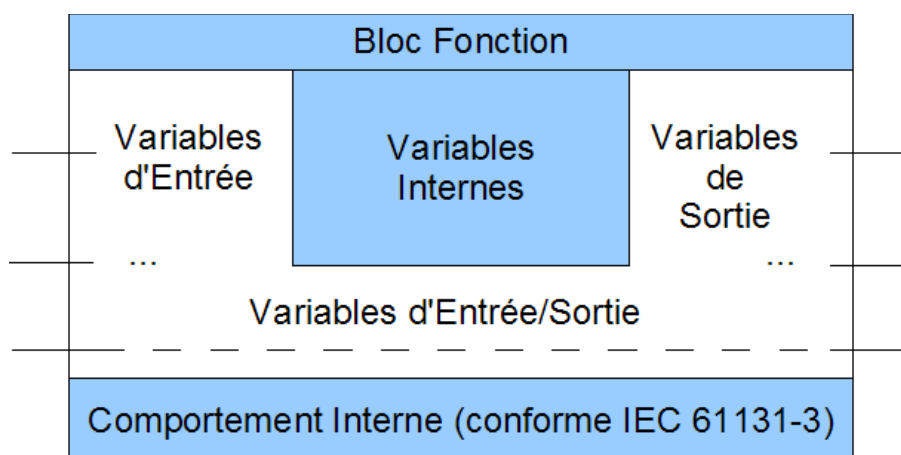


FIGURE 3.9 – Vue Générale d'un Bloc Fonction (**FB**) [Chiron 09]

Les **FB** sont très utiles car ils permettent l'abstraction de logiques complexes et récurrentes au sein d'un programme, augmentant l'efficacité et la rapidité de mise en œuvre de fonctionnalités présentes à plusieurs emplacements dans le programme, par exemple une boucle de régulation de température, un régulateur Proportionnel Intégral Dérivé (PID). Outre le gain de mémoire obtenu en multipliant des instances de variables au lieu de retrouver la même portion de code plusieurs fois, ils permettent également de protéger des parties de code automate en les rendant inaccessibles à un utilisateur non autorisé à modifier le bloc, ce qui est utile lorsque les machines sont fournies avec un code accessible par le service maintenance d'une entreprise cliente. Fabien Chiron [Chiron 09] utilise ce bloc lors du déploiement des spécifications avec l'extension **SysML**.

La représentation des **FB** avec **UML** 2.0 a fait l'objet de travaux, par exemple Heverhagen [Heverhagen 01] qui introduit une représentation personnalisée qui se présente sous une forme hybride entre les blocs fonctions et les classes **UML** 2.0. Bien que difficilement exploitable directement, le travail de Heverhagen soulignait de façon pertinente les insuffisances graphiques du langage dans le cadre de la représentation d'éléments possédant des connexions statiques de variables et il est fortement probable que ses travaux aient influencé le concept de bloc en **SysML**.

Dans [Chiron 07], il est proposé d'exploiter directement les **FB** avec **SysML** qui est facilement compatible avec la norme IEC 61131-3. En 2005, un projet de conception de Functions Blocks dédié aux holons a été développé sous le nom de HoloBlock³ pour faciliter le dé-

3. <http://www.holobloc.com> (accessible au 27 octobre 2010)

ploiement d'un holon. [Mařík 02] nous explique qu'un FB, bien qu'adapté pour le contrôle en temps réel de la partie matériel d'un holon, est limité pour le contrôle d'échange d'informations complexes. Ils donnent aussi un exemple d'utilisation de FB pour le contrôle d'un Système holonique d'Assemblage en expliquant que les FB permettent d'être plus flexible et le contrôle temps-réel est mieux respecté avec son utilisation qu'avec l'utilisation de la technologie multi-agent. Les agents posent aussi le problème de la réutilisabilité pour des environnements différents. Ils décomposent un holon en plusieurs FB avec des rôles différents. Ils rejoignent l'utilisation faite par [Chiron 07] qui décompose un composant en plusieurs facettes, chaque facette étant composée de FB.

3.5.2 Le langage UML et ses extensions

L'approche objet a donné naissance à beaucoup de méthodes et langages de modélisation. Le consortium *Object Management Group* (OMG)⁴ s'en est inspiré pour proposer un langage à but universel : *Unified Modeling Language - Langage de modélisation unifié* (UML). UML ne propose pas de méthode mais seulement une liste de diagrammes composés en deux types : « structurel » et « comportemental » [OMG 10b]. Les diagrammes structurels montrent la structure statique des objets dans un système ; c'est-à-dire qu'ils décrivent les éléments sans tenir compte du temps. Les diagrammes comportementaux montrent le comportement dynamique des objets d'un système, incluant leurs méthodes, leurs collaborations, leurs activités et leurs historiques des états. Le comportement dynamique d'un système peut être décrit comme une série de changements dans le système à travers le temps. À cette norme, il est possible d'ajouter des profils et des stéréotypes qui permettent de l'adapter à une classe de système. Elle donne lieu à plusieurs extensions.

3.5.2.1 AUML

Une extension du standard UML appelée *Agent Unified Modeling Language* (AUML) a été proposée dès le début des travaux sur les SMA. L'avantage de cette approche est qu'elle est basée sur UML, destinée à la programmation orientée-objet et très bien connue dans le domaine du génie logiciel, ce qui permet aux développeurs de passer facilement d'une méthodologie orientée-objet vers l'approche orientée-agent. AUML a été proposée par [Bauer 01], qui a suggéré des nouvelles spécifications et de nouveaux diagrammes pour pouvoir présenter les interactions entre les agents dans les SMA. Elle a été utilisée par exemple récemment dans les travaux de [Ali 09] sur le contrôle du trafic urbain. Ce projet est actuellement ralenti car la version 2.1. d'UML et son extension SysML ont pris en compte des éléments développés dans AUML⁵.

3.5.2.2 UEML

Unified Enterprise Modeling Language - langage unifié de modélisation d'entreprise (UEML) est le résultat du projet UEML [Petit 02] dont l'objectif principal était de résoudre le problème de l'existence d'une multitude de langages de modélisation d'entreprise. La disparité des langages et méthodes de modélisation d'entreprise impose la définition d'un langage commun ou générique permettant aux modèles de communiquer. Cette fusion devrait permettre

4. <http://www.omg.org> (accessible au 27 octobre 2010)

5. <http://www.auml.org> (accessible au 27 octobre 2010)

l'émergence d'un langage unifié dont la couverture finale sera supérieure à celle assurée individuellement par chacun des langages actuels. A terme, **UEML** devrait conduire à assurer l'indépendance et l'interopérabilité entre chacun des formalismes initiaux, permettant ainsi de couvrir davantage les problématiques relevant de la modélisation d'entreprise [Vallespir 03].

3.5.2.3 SysML

Systems Modeling Language - Langage de Modélisation des systèmes (**SysML**) est un profil d'**UML2** qui étend son champ d'application initial, le génie informatique, à l'ingénierie des systèmes. Son objectif est de supporter la spécification, l'analyse, la conception, la vérification et la validation d'une grande quantité de systèmes complexes. Ces systèmes peuvent inclure des systèmes physiques, des logiciels, des systèmes d'information, des systèmes automatisés, des processus, des personnes et des biens [OMG 10a].

Elle introduit un nouveau stéréotype nommé « Block ». Pour le représenter, **SysML** introduit un nouveau type de diagramme, le *Block Definition Diagram - diagramme de définition de bloc* (**BDD**). De plus, il permet également de décrire la structure interne d'un bloc avec le *Internal Block Diagram - diagramme interne de bloc* (**IBD**) de la même façon que dans les diagrammes de structure classique UML 2.0. Le stéréotype bloc hérite directement d'une classe **UML** 2.0 mais a la possibilité d'accueillir un nouveau type de port, les ports de flux, traduction de « flow ports ». **SysML** a aussi ajouté aux diagrammes du langage **UML**, un diagramme pour pouvoir représenter simplement des exigences sur un système [Evrot 08].

3.5.2.4 UML pour le Temps Réel

Modeling and Analysis of Real-Time and Embedded systems (**MARTE**)⁶ est un profil **UML** pour la gestion des systèmes temps réel et embarqués [Gérard 07]. Il rajoute les notions de NFP (Non-Functional Properties) comme par exemple la gestion des contraintes de performance, le taux d'utilisation de la mémoire ou la consommation... Ces contraintes sont écrites avec un langage proche de l'OCL présenté dans **UML**. **MARTE** est inspiré d'un profil **UML** plus ancien : SPT (Schedulability Performance and Time). Il ajoute des expressions pour le temps (after, when...). Il est capable de gérer le temps discret et le temps continu. Certains éléments de **MARTE** sont empruntés à **SysML**.

3.5.3 LOTOS

Language Of Temporal Ordering Specification (**LOTOS**) est un langage permettant de décrire formellement les systèmes contraints par le temps. Il s'agit d'un langage de type algèbre de processus. **LOTOS** [Bolognesi 87] est une Technique de Description Formelle pour la spécification et la conception de systèmes de traitement distribués. Une spécification **LOTOS** se présente sous la forme d'un processus structuré en d'autres processus. Un processus **LOTOS** est une boîte noire qui communique avec son environnement au travers de portes et sur le principe d'une offre de rendez-vous. Le parallélisme et la synchronisation entre processus s'exprime par des opérateurs de composition : mise en séquence, synchronisation sur toutes les portes, synchronisation sur certaines portes, choix non déterministe et entrelacement (composition parallèle sans synchronisation).

RT-LOTOS (Real-Time **LOTOS**) [Courtiat 00] étend le langage **LOTOS** par des opérateurs temporels (délai, latence, offre limitée dans le temps). La combinaison d'un délai détermi-

6. <http://www.omgmarTE.org> (accessible au 21 octobre 2010)

niste et d'un délai non déterministe permet de traiter les intervalles temporels. Le langage ainsi étendu conserve la partie « contrôle » de **LOTOS**, mais remplace les types de données algébriques par des implantations en C++ ou Java. D'autres améliorations de ce langage sont développées comme par exemple **TLSA** (Time Labeled Scheduling Automata) [Lohr 02] qui permet l'amélioration du concept d'automates à état temporisé en réduisant les graphes (évite l'explosion combinatoire).

3.5.4 L'analyse de déroulement, la norme JIS Z 8206

Ce formalisme présente les avantages d'être très répandu dans l'industrie, d'utiliser des symboles simples à mémoriser et d'un niveau de détail pas trop important, de mettre en évidence les actions qui ne correspondent pas à de la valeur ajoutée (toutes celles qui ne sont pas des opérations). On obtient un modèle graphique synthétique et général des flux physiques. Parmi les inconvénients, on peut justement regretter de ne pas voir apparaître les flux informationnels. De plus, son utilisation ne permet pas la gestion de flux simultanés sans la mise en place de règles de gestion. Ce standard a été développé par [Ishiwata 97] et a été utilisé dans les travaux de [Lee 99] et récemment dans [Fontanili 06]. Il utilise cette norme pour la modélisation de la vue ressource d'une analyse qualité dans un laboratoire pharmaceutique. Des matrices et diagrammes basés sur des symboles standardisés décrivent le flux du processus et permettent de déterminer rapidement où les problèmes peuvent se poser dans le processus. Ce flux peut inclure des étapes séquentielles d'opération de production, ou il peut inclure les étapes séquentielles que des opérateurs humains peuvent avoir à réaliser. Les symboles sont divisés en symboles basiques et supplémentaires. Les graphiques basiques consistent en six symboles qui sont l'opération d'addition, le transport, le stockage, la temporisation, l'inspection du volume et l'inspection de la qualité (table 3.3).

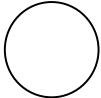
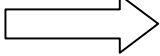




Opération	Transport	Stockage	Temporisation	Contrôle de la qualité	Contrôle du volume
					

TABLE 3.3 – Signification des symboles de la norme JIS Z 8206

Le diagramme de déroulement complète la matrice de déroulement. Il est utilisé pour étudier chaque étape en dessinant la mise en place de chaque zone dans laquelle le processus se déroule. Cette mise en place sera ainsi compréhensible plus facilement en utilisant les mêmes symboles utilisés que dans la matrice de déroulement qui fait office de fiche technique. Ceci est utilisé par exemple dans [Lee 99], où on peut voir une figure qui illustre un diagramme de déroulement qui décrit le chemin, la séquence et les distances pour un processus. En utilisant les matrices et les diagrammes de déroulement, les problèmes peuvent être identifiés facilement. Changer la séquence des étapes, éliminer ou ajouter une étape, ou changer la localisation des étapes dans le processus peut être fait, simplement, pour améliorer le processus.

3.5.5 Formalisme pour l'analyse d'un flux de production

[Bourrières 90] utilise quatre symboles de base pour la réalisation d'un schéma opérationnel (table 3.4). La figure 3.10 montre l'utilisation de ce formalisme pour représenter un processus d'assemblage.

Opération fonctionnelle unitaire	Opération fonctionnelle binaire	Opération positionnelle	Opération auxiliaire de désassemblage

TABLE 3.4 – Les quatre symboles de base de [Bourrières 90]

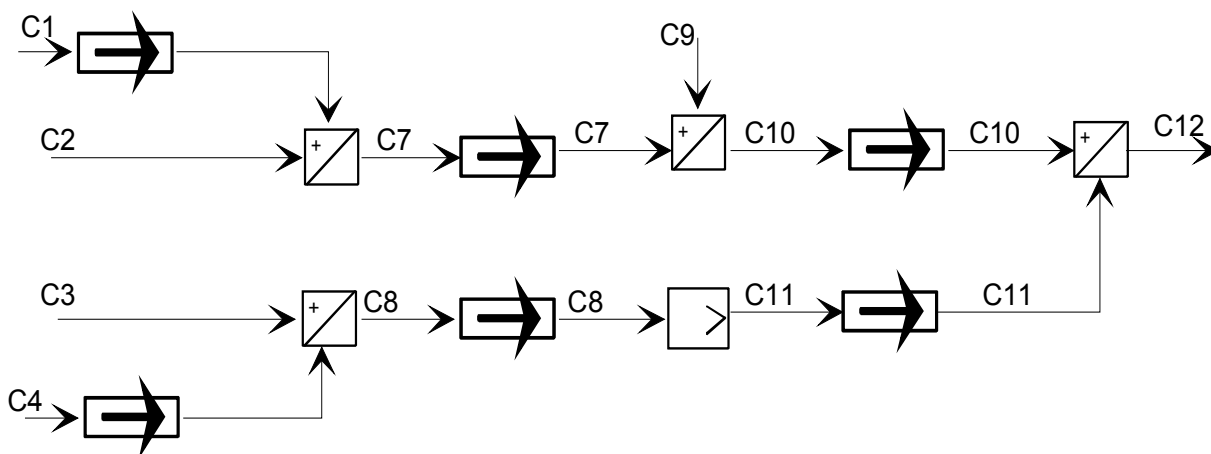


FIGURE 3.10 – Exemple d'un processus d'assemblage [Fontanili 99]

A partir de ce formalisme, [Paris 93] propose une extension destinée aux lignes d'assemblage automatisées, avec un système de convoyeurs. La figure 3.11 présente les principaux symboles de ce formalisme, et la figure 3.12 représente un exemple d'application à un système d'assemblage.

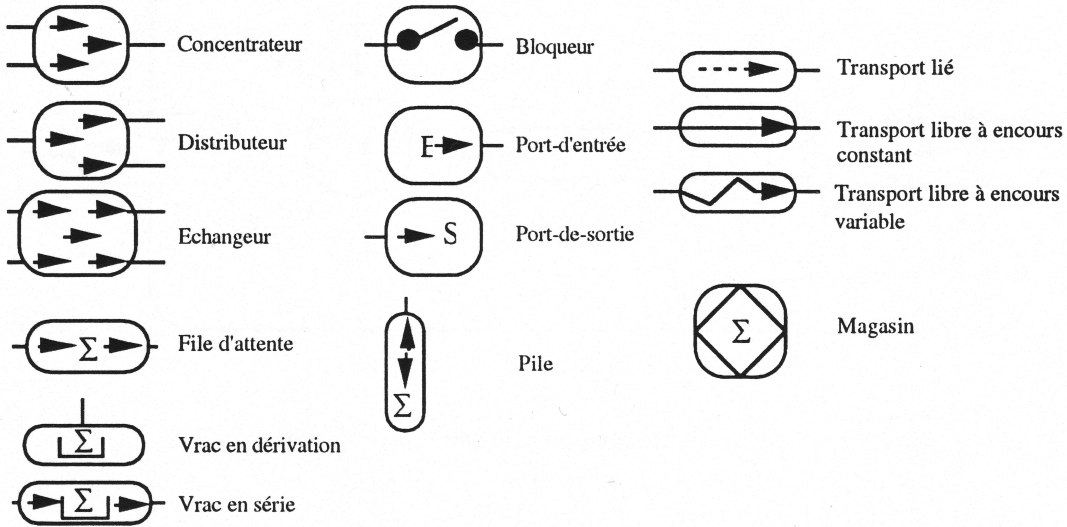


FIGURE 3.11 – Bibliothèque des différentes entités transitiques de [Paris 93]

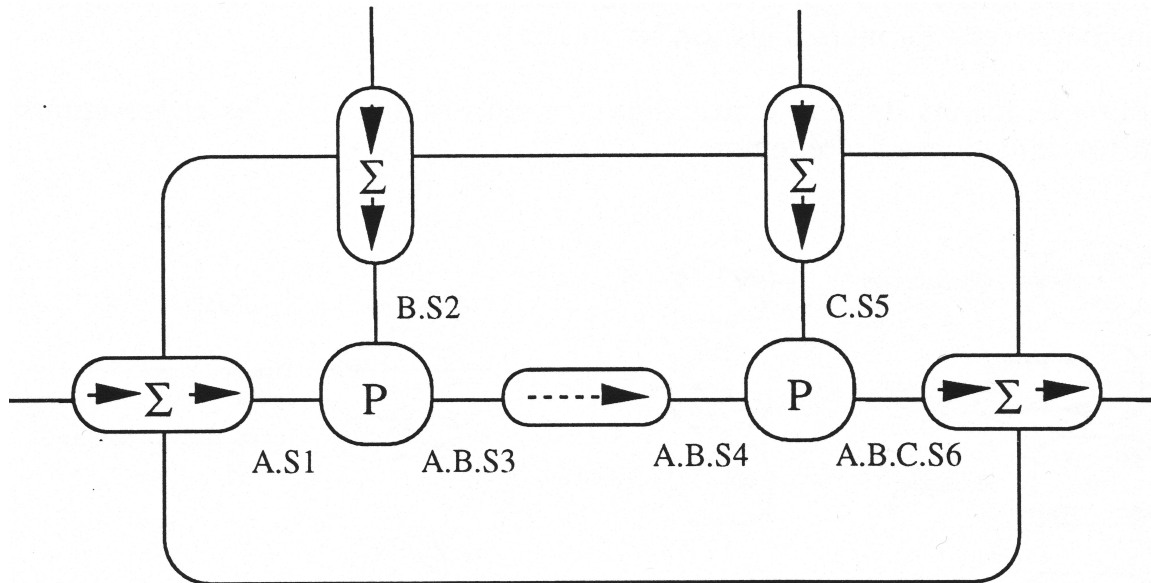


FIGURE 3.12 – Exemple d'application avec une cellule composée de deux postes simples [Paris 93]

Ce formalisme est d'une précision beaucoup plus importante que celui utilisé pour l'analyse de déroulement. D'autre part, il est d'une compréhension facile et d'une similitude assez proche avec les éléments réels qui composent un système de production. On dispose de beaucoup de détails sur le sous-système physique, mais par contre on n'a aucune information sur

les sous-systèmes logiques et décisionnels. Son inconvénient par rapport à l'analyse de déroulement est que, bien que plus précis, il est aussi plus complexe et donc moins adapté à la communication lors de réunion avec les différents acteurs.

3.5.6 Les réseaux de Petri

Parmi les autres formalismes rencontrés dans la littérature concernant la formalisation d'un processus, il y a les *Réseaux de Petri* (RdP) [David 92] pour lesquels les travaux et applications sont très nombreux. Le fonctionnement des RdP est basé sur une dynamique qui autorise le franchissement des transitions sous certaines conditions de présence de jetons dans les places d'entrée. Son avantage est de proposer des outils mathématiques performants pour représenter et analyser un système. Le reproche que l'on peut lui faire pour ce type d'utilisation est d'être d'un niveau d'abstraction trop important, notamment pour répondre à l'objectif de communication entre les différents acteurs.

Le GRAFCET est une normalisation industrielle d'une façon d'interpréter (actions et conditions) proche des réseaux de Petri [IEC60848 88]. Elle est très utilisée pour la programmation d'API.

3.6 UTILISATION DES MODÈLES

Tous ces outils ont pour objectif le développement de modèles graphiques. Les modèles ainsi développés sont utilisés pour deux raisons principales :

- Pour permettre une communication aisée avec les acteurs du système à modéliser, c'est-à-dire servir de document de travail sur un projet. Les modèles doivent être, en plus d'être rigoureux et précis, facilement compréhensibles par des non-informaticiens,
- Pour permettre par la suite une écriture de code facilitée dans différents langages (langages de programmation informatique ou d'automate, langages de simulation. . .).

Le choix entre les différents outils devra être un compromis entre ces deux utilisations. Ces outils devront participer à la discussion avec les acteurs du système pour pouvoir par la suite permettre facilement de coder en ayant compris les enjeux et les idées que ces acteurs ont voulu transmettre. Par exemple sur le domaine du *Circuit Du Médicament* (CDM), les acteurs du système seront les pharmaciens, les préparateurs et les managers du CHU. Les outils utilisés devront être compréhensibles par les acteurs lors des réunions que nous aurons avec eux. Cependant, ils doivent aussi nous permettre de concevoir le plus aisément possible les éléments qui nous seront demandés.

3.7 CONCLUSION

Nous nous sommes intéressés dans ce chapitre aux architectures, méthodologies et méthodes de modélisation des systèmes de production de biens et de services. Nous avons présenté les cadres d'architecture d'une entreprise, puis les méthodologies pour la conception de systèmes complexes dont les *Systèmes Multi-Agent* (SMA). Nous avons aussi exposé les méthodes les plus usitées ainsi que des langages de modélisation pour modéliser les flux de production ou pour permettre une analyse et une spécification de système complexe.

On peut encore citer quelques travaux qui ont développé des environnements en tentant d'utiliser les avantages de chaque outil comme par exemple GERAM (Generalised Enterprise

Reference Architecture and Methodology) [GERAM 99] ou GME (Generic Modeling Environment) [GME 05, Aloui 07].

Nous avons remarqué ainsi la tendance actuelle d'aller vers l'utilisation de l'approche objet. En effet, les méthodes orientées objet permettent de mieux s'adapter aux changements du contexte de réalisation ou de l'évolution des besoins au cours du temps [Mertins 97]. Nous avons remarqué aussi que l'approche objet tend elle-même vers la conception par composants réutilisables. Une autre tendance forte est l'utilisation de l'Ingénierie Dirigée par les Modèles (IDM) pour une spécification la plus tardive possible grâce à l'utilisation de différents modèles sur plusieurs niveaux d'abstraction. Cette spécification tardive permet de s'adapter à différentes technologies suivant le contexte d'utilisation des modèles.

Dans ce chapitre, nous avons recensé les principaux outils de modélisation des systèmes de production de biens et de services. Nous avons vu dans le chapitre 1 que le *Circuit Du Médicament* (CDM) était un système holonique, nous allons voir, dans le chapitre suivant, quels sont les outils de modélisation recensés dans ce chapitre et qui ont été appliqués à de tels systèmes, puis plus particulièrement aux *Systèmes Contrôlés par le produit* (SCP).

4

DE LA MAÎTRISE D'UN SYSTÈME HOLONIQUE

RÉSUMÉ DU CHAPITRE 4

Nous montrons dans ce chapitre les différents travaux qui nous ont semblés les plus importants concernant la maîtrise d'un système holonique. Une première partie présente des travaux sur la modélisation d'un système holonique. Elle fait un bilan des outils présentés dans le chapitre précédent pour une application dédiée aux systèmes holoniques. Une deuxième présente ceux plus particulièrement sur les systèmes holoniques appliqués aux Systèmes Contrôlés par le Produit. L'objectif est de pouvoir faire notre choix pour pouvoir proposer une méthodologie de réingénierie pour les systèmes holoniques dans le chapitre suivant.

SOMMAIRE

4.1	INTRODUCTION	64
4.2	MODÉLISATION D'UN SYSTÈME HOLONIQUE	64
4.3	MAÎTRISE D'UN SYSTÈME HOLONIQUE VIA LE CONTRÔLE PAR LE PRODUIT	67
4.4	CONCLUSION	70

4.1 INTRODUCTION

Après avoir présenté les principaux outils de modélisation dans le chapitre précédent et les principaux concepts consécutifs au paradigme holonique dans le chapitre 2, nous faisons le lien entre ces deux chapitres. En effet, une nouvelle approche en productique s'est faite jour dans les années 90, visant à aborder la gestion et la conduite d'un système de production en accordant plus d'importance au produit lui-même. Une approche de pilotage centrée sur le produit (« product-centric ») a connu un développement dans le cadre du paradigme holonique.

Nous présenterons quelques travaux qui ont déjà modélisé un système holonique et les différentes architectures, méthodologies, méthodes et langages qu'ils ont utilisés. La seconde section de ce chapitre présente quelques travaux sur la modélisation de systèmes holoniques en général, la troisième section est centrée sur les travaux sur les systèmes holoniques consacrés aux *Systèmes Contrôlés par le produit (SCP)*.

4.2 MODÉLISATION D'UN SYSTÈME HOLONIQUE

Une des modélisations les plus citées sur le contrôle par les ressources des systèmes manufacturiers holoniques est l'architecture **ADACOR** [Leitão 06]. La figure 4.1 représente le lien entre les différentes ressources à travers le réseau de l'entreprise.

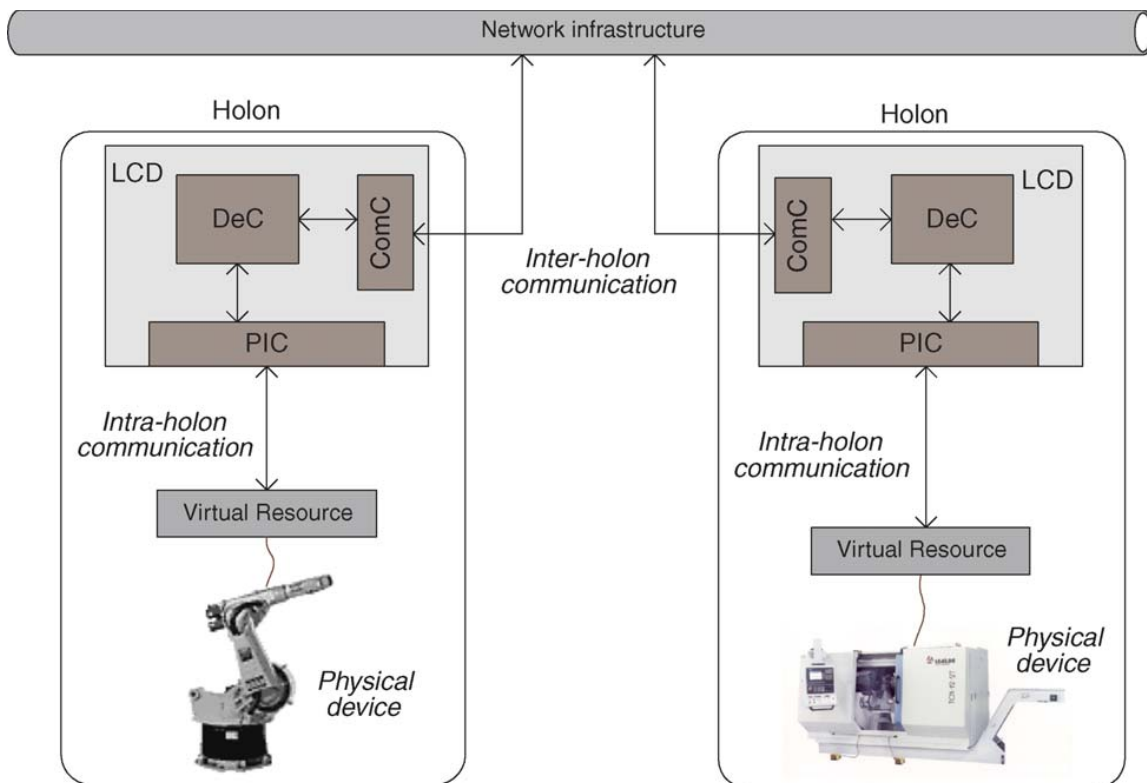


FIGURE 4.1 – Modèle conceptuel pour les holons **ADACOR** [Leitão 06]

L'auteur propose une modélisation basée sur les ressources, la communication entre les différents holons est exprimée avec des diagrammes d'interaction **AUML**, qui sont une extension des diagrammes de séquence d'**UML** pour les **SMA**. On peut voir un exemple simple de

communication entre un Ressource-Holon et un Task-Holon sur la figure 4.2. Les Task-holons permettent de gérer la communication et les choix de pilotage entre chaque ressource.

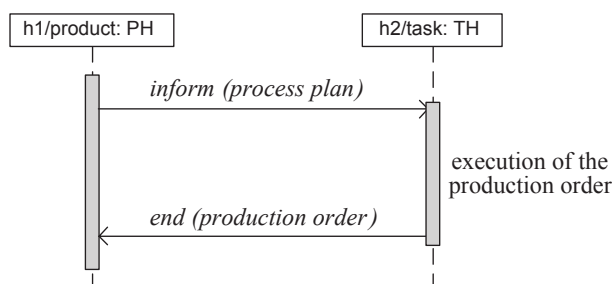


FIGURE 4.2 – Interaction entre les Product Holon et les Task Holons avec un diagramme d'interaction AUMML [Leitão 04]

Le comportement dynamique de chaque classe d'holon est modélisé formellement en utilisant les Réseaux de Petri (RdP). Ils modélisent le comportement de la production et du pilotage [Leitão 04].

Le laboratoire de Marseille, le LSIS, a effectué plusieurs travaux en utilisant l'architecture PROSA et le contrôle par les ressources. On peut citer par exemple [Demongodin 08] qui utilise les RdP pour permettre une interaction entre les holons produits et ressources.

[Blanc 08] propose un MES, en utilisant le concept de HMS qu'il illustre avec un système manufacturier qui produit des vitres pour les automobiles. Cette structure est basée sur une typologie d'éléments composant un système manufacturier (produits, ressources, ordres) basés sur PROSA. Ils voient le Holon-Ordre (HO) comme la partie logique du produit. Ils proposent aussi une façon d'ordonner les Holon-Ressource (HR) pour permettre un maximum de flexibilité et pour centrer le système sur le produit. Ils utilisent un diagramme de classe UML pour montrer les relations entre les différents holons. Selon les auteurs, ce pilotage est centré sur le produit à cause des événements et des activités qui sont gérés par les HO sauf pour l'ordonnement et les patrons de conception qui sont gérés par les HR. La création des HO correspond aux exigences d'un produit, cet HO est en charge de solliciter les services et contracter les HR. Le cycle de vie d'un HO est montré avec des diagrammes UML d'état-transition.

[Simão 05] et [Baïna 06a] utilisent un méta-modèle des systèmes holoniques pour définir les relations entre chaque élément. [Simão 05] centre sa modélisation principalement sur les HR pour un système pouvant être dirigé par des règles. Il utilise aussi l'architecture PROSA. Il définit des stéréotypes pour pouvoir utiliser les méta-classes : « Holonic Class » et « PROSA HOLON » pour simplifier ses modèles, ils sont faits sous UML. Il propose ensuite des règles qui permettront à travers ces HR de rendre le système centré sur le produit. Il propose plusieurs modèles pour définir un HMS et les relations entre holons. Le modèle de [Simão 05] représenté sur la figure 4.3 présente un HMS et les holons définis dans PROSA. Le modèle représenté sur la figure 4.4 représente un HR avec ces liens d'héritage. Il explique qu'un HR peut être une station de travail, un magasin ou un équipement. Une station de travail peut être, par exemple, une station d'assemblage ou une station de fraisage. Une station d'assemblage est par conséquent un HR du système.

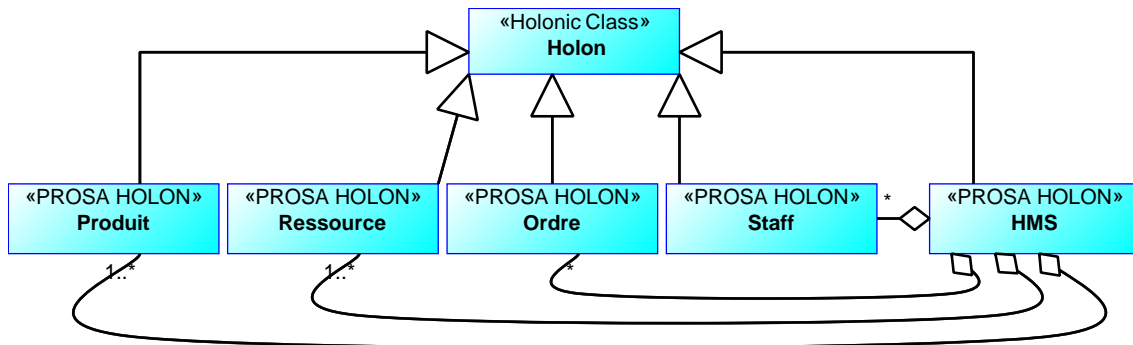


FIGURE 4.3 – Les principales « Holonic class » d'après [Simão 05]

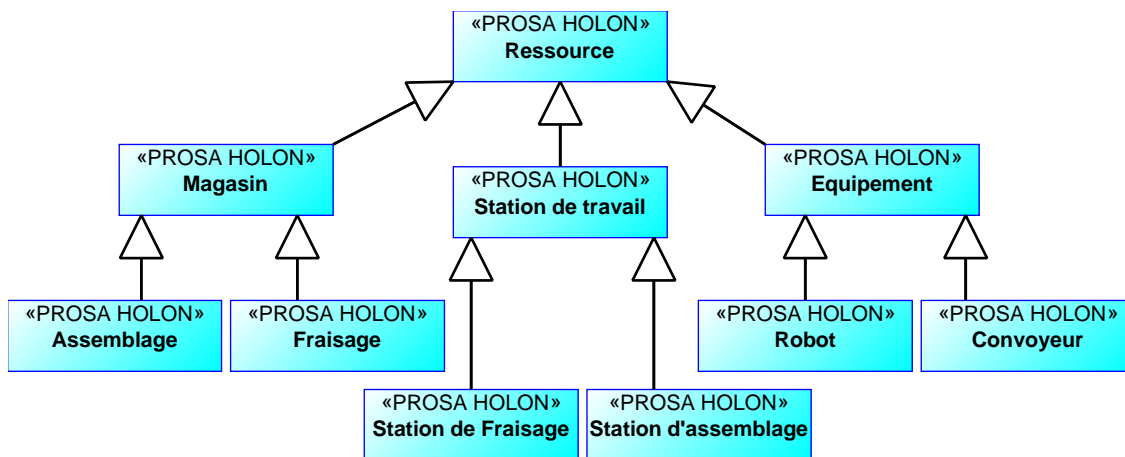


FIGURE 4.4 – Spécialisation des holon-ressources d'après [Simão 05]

4.3 MAÎTRISE D'UN SYSTÈME HOLONIQUE VIA LE CONTRÔLE PAR LE PRODUIT

Des auteurs qui ont travaillé sur le paradigme **SCP** utilisent le paradigme holonique pour pouvoir le contrôler. Dans le paradigme de « Système contrôlé par le Produit », il s'agit de conférer au produit un rôle actif et participatif dans les décisions du système.

[Auzelle 08] a cherché à modéliser un système d'information centré sur le produit dans une entreprise. [Auzelle 09] s'est inspiré du concept holonique et de **SdS** pour formaliser un « Système-Entreprise » en une sorte de « Système de Systèmes d'Information » (SdSI) en tenant compte de ses aspects récurrents et multi-échelles dans un contexte organisationnel évolutif. Parmi tous les points de vue d'un système, cette représentation lui permet de représenter ceux du « Système A Faire » et ceux du « Système Pour Faire » avec son contenu d'Ingénierie associée guidée par le cadre de modélisation de Zachman. Il a outillé cette proposition de cadre d'Ingénierie Système Basée sur des Modèles (ISBM) en étendant le méta-modèle de l'outil « MEGA Modelling Suite » ainsi qu'en développant les interfaces utilisateurs nécessaires et il l'a appliqué à l'Ingénierie d'un système de traçabilité d'un Produit.

Plusieurs auteurs préconisent l'utilisation de la démarche **MDA** (cf section 3.2.4) par une abstraction à 4 niveaux [Bézivin 01, Naumenko 03, Bézivin 05]. Le niveau le plus bas Mo représente les différents objets de modélisation aussi appelés « univers du discours ». Le niveau M1 représente les différents modèles de chaque univers de discours. Les modèles du niveau M1 appartiennent à différents domaines d'intérêt relatifs aux univers du discours représentés par les modèles. Il est possible qu'un domaine d'intérêt recouvre plusieurs univers du discours. De même, des modèles provenant de différents univers de discours peuvent relever du même domaine d'intérêt.

Définition 4.1 *Un méta-modèle est un modèle de spécification pour une classe du système à étudier où chaque système à étudier dans la classe est lui-même un modèle valide exprimé dans un certain langage de modélisation [Kleppe 03].*

Le niveau M2 représente les méta-modèles (définition 4.1) spécifiques à chaque domaine : un méta-modèle pour chaque domaine d'intérêt pertinent pour les modèles de niveau M1. Finalement, le niveau M3 présente le méta-méta-modèle et doit être conçu pour permettre la définition de tous les concepts nécessaires pour la modélisation de méta-modèles et pour leur unification dans un cadre de référence commun. Un méta-méta-modèle est indépendant du domaine, il contient des méta-caractéristiques pour des méta-modèles spécifiques [Baïna 06b]. Ces différents niveaux sont résumés sur la figure 4.5.

[Baïna 06b] a proposé de faire le lien entre un **ERP** et un **MES** par un méta-modèle basé sur le produit (processus holonique) qui est vue comme un holon (figure 4.6). Il est proposé d'adapter le concept d'holon pour résoudre le problème de synchronisation entre des vues informationnelles et physiques d'un même objet. Les auteurs proposent une approche dirigée par les modèles, basé sur le **MDA**, pour résoudre le problème d'interopérabilité des modèles holoniques. Un méta-modèle d'un système holonique a été construit en langage **UML**. Puis, une comparaison a été faite avec le méta-modèle d'**UEML** et l'environnement MEGA en mettant en place des règles de transformation entre méta-modèles. Elle utilise aussi le cadre de Zachman pour cibler son processus de modélisation. [El Haouzi 08a] propose une approche méthodologique générale pour la conception de **SCP** en 5 étapes. Dans l'étape de « Design », l'étape la plus importante, elle propose aussi une division en 4 niveaux ontologiques.

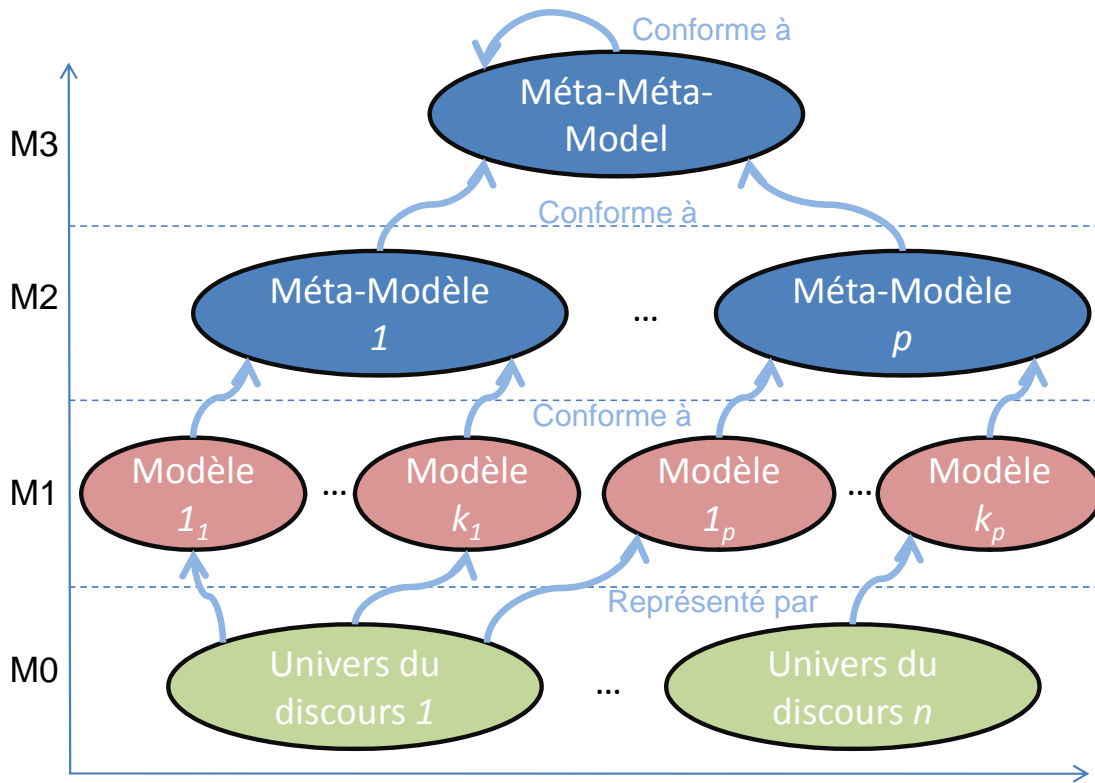


FIGURE 4.5 – L'approche par quatre niveaux ontologiques

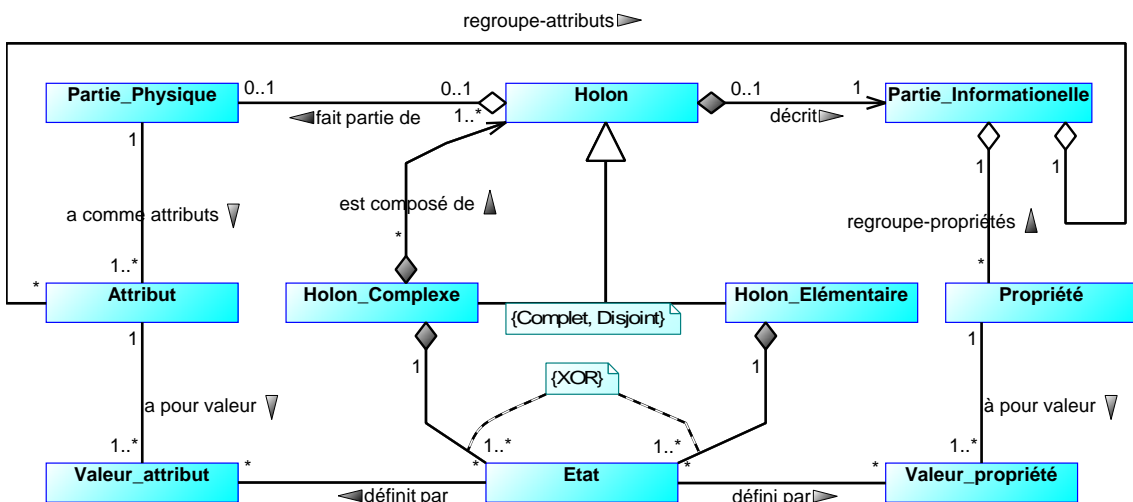


FIGURE 4.6 – Méta-modèle holonique pour la représentation du produit d'après [Baïna 06b]

Elle utilise la méthodologie **ASCI** (cf section 3.3.2) pour faire un modèle de connaissance générique d'un système d'assemblage centré sur le produit, cela correspond à son niveau le plus abstrait (niveau N3 de la figure 4.7).

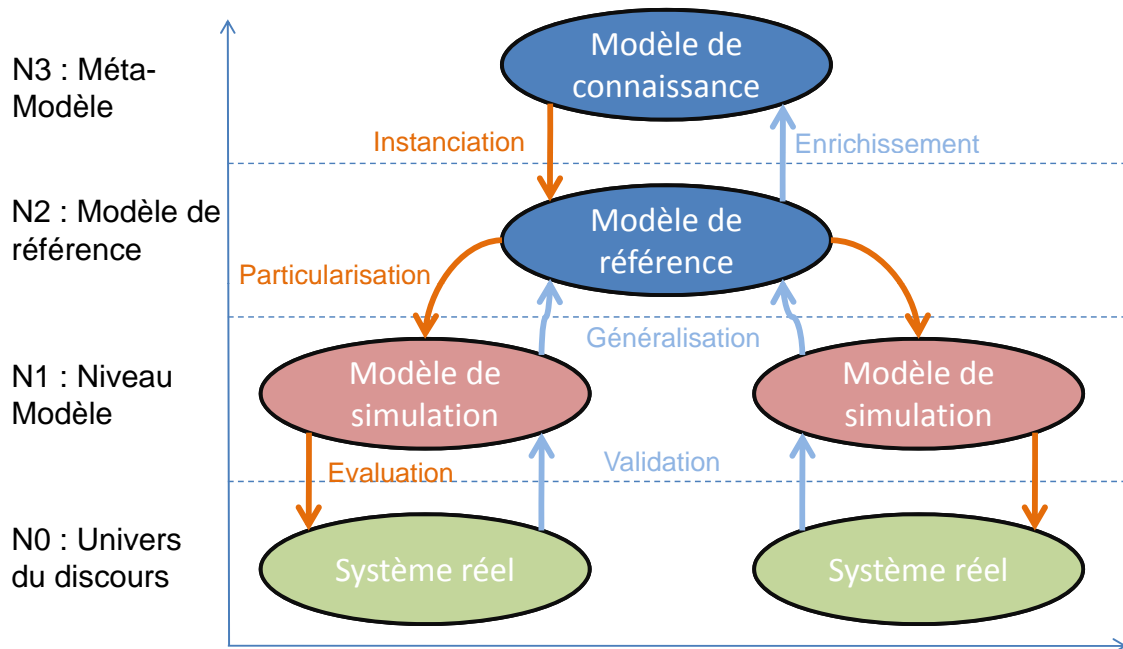


FIGURE 4.7 – Les différents niveaux ontologiques utilisés par [El Haouzi 08a]

Le modèle de référence est conçu après une instanciation de ce modèle de connaissance (niveau N2). Ce modèle de référence est après mis en place avec un modèle de simulation (niveau N1) qui communique avec le système réel qui correspond à un univers de discours (niveau N0). Le modèle de connaissance est enrichi dans la phase d'Analyse par rapport à la version originelle d'**ASCI** par un diagramme représentant les entités décisionnelles **ED** en plus des trois sous-systèmes classiques (physique, logique et décisionnel) [El Haouzi 08b]. Il permet de faire le lien entre le sous-système physique et décisionnel (figure 4.8). Les différents diagrammes sont exprimés avec le langage **UML**. [Gouyon 07] s'est consacré à l'étude du contrôle des ressources par le produit principalement avec l'environnement MEGA. Il présente dans un deuxième temps, une méthode avec automates à états finis pour la gestion de la production des produits. [Cardin 07] utilise les **RdP** pour gérer correctement l'avance et le retard d'observateurs lors de la simulation en ligne d'un **SCP**.

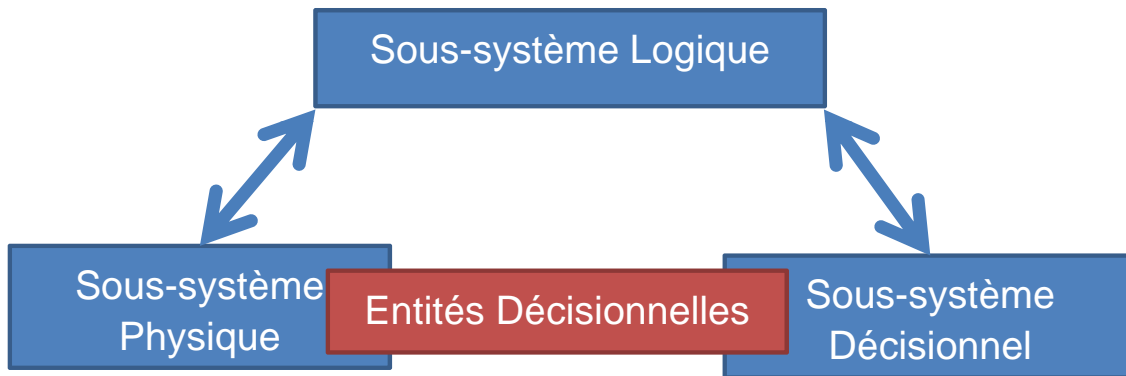


FIGURE 4.8 – Enrichissement du lien entre les différents sous-systèmes dans la phase d'Analyse d'ASCI

4.4 CONCLUSION

Nous avons passé en revue dans ce chapitre quelques travaux sur les systèmes holoniques. En premier lieu, nous avons présenté plusieurs façons de modéliser des systèmes holoniques en général. Ensuite, nous avons centré notre attention sur les systèmes dont le contrôle est effectué par le produit.

Nous avons vu que le fonctionnement le plus sécurisé du *Circuit Du Médicament* (CDM) correspond à un *Système Contrôlé par le Produit* (SCP) dans le chapitre 1, nous avons alors exposé les différents concepts utilisés pour modéliser un SCP (chapitre 2), nous avons fait ensuite un état de l'art des principaux outils de modélisation (chapitre 3) et nous avons étudié dans ce chapitre ceux qui ont été appliqués aux systèmes holoniques. Ces travaux sont intéressants mais aucun d'entre eux, à notre connaissance, ne propose une méthodologie pour la réingénierie. Cependant, nous pourrions nous inspirer des différentes démarches exposées dans ce chapitre, comme par exemple la transformation de modèle ou la gestion de l'interopérabilité dans une entreprise, pour en proposer une, ce que nous allons faire dans le chapitre suivant.

Nous avons maintenant tous les éléments en notre possession pour proposer une méthodologie complète de réingénierie d'un système holonique avec comme objectif une réorganisation en SCP. Ensuite, à partir de cette méthodologie, nous pourrions l'appliquer à la réingénierie du CDM pour pouvoir le réorganiser de façon plus sécurisée.

Notre principal objectif sera donc de permettre la réingénierie pour un contrôle par le produit en faisant le lien entre les travaux sur le pilotage de l'atelier [Gouyon 07, Cardin 07] et sur l'interopérabilité entre les SI d'une entreprise [Auzelle 08].

Notre contribution méthodologique est présentée dans le chapitre suivant.

5

NOTRE APPROCHE MÉTHODOLOGIQUE : RÉINGÉNIERIE POUR LE CONTRÔLE PAR LE PRODUIT DE SYSTÈMES MANUFACTURIERS

RÉSUMÉ DU CHAPITRE 5

Ce chapitre présente notre contribution méthodologique. Dans un premier temps, nous présentons et expliquons nos choix d'outils de modélisation. Nous utilisons une démarche MDA à quatre niveaux d'abstraction :

- 1. Le premier niveau présentera le méta-méta-modèle, c'est-à-dire la description de toutes les entités et concepts qui seront utilisés dans notre approche,*
- 2. Le second niveau présentera les méta-modèles ; nous divisons ce niveau en deux parties, une représentant un système manufacturier holonique non-contrôlé par le produit puis une deuxième représentant le lien entre les holons lors d'un contrôle par le produit,*
- 3. Le troisième niveau sera le Modèle Générique de Connaissance d'un domaine, le modèle sera composé d'une instanciation des méta-modèles,*
- 4. Le dernier niveau représentera un Modèle de Connaissance d'un système du domaine et sera une instanciation du niveau supérieur.*

La démarche MDA est complétée par une méthodologie ASCI modifiée (ASCI-RHMS) pour faire le lien entre les trois derniers niveaux. ASCI est adaptée pour concevoir un Modèle Générique de Connaissance d'un domaine et un Modèle de Connaissance d'un système, cependant certains éléments doivent lui être ajoutés pour qu'elle puisse être compatible avec la démarche MDA. Les langages employés sont UML et SysML, SysML étant une extension d'UML adaptée à la modélisation de systèmes composés de parties logicielles et matérielles. L'approche utilisée dans ce chapitre est l'approche par composant, en utilisant ASCI-RHMS ; nous l'appliquons pour construire une bibliothèque de composants réutilisables. Nous finissons ce chapitre en expliquant les différentes utilisations de cette méthodologie, soit en partant du niveau du méta-modèle pour construire un Modèle Générique de Connaissance d'un Domaine, soit en partant de ce Modèle Générique en l'instanciant pour un système particulier du domaine, soit en faisant une étude complète d'un système avec son domaine en reprenant les deux utilisations précédentes.

SOMMAIRE

5.1	INTRODUCTION	73
5.2	CHOIX DES OUTILS DE MODÉLISATION	73
5.2.1	Choix de l'approche	73
5.2.2	Choix de l'architecture	74
5.2.3	Choix du langage	76
5.3	VOCABULAIRE EMPLOYÉ	77
5.3.1	Mise au point sur le terme « générique »	77
5.3.2	Transformation entre niveaux d'abstraction	77
5.4	LE NIVEAU M ₃ : DESCRIPTION D'UN SYSTÈME HOLONIQUE DE PRODUCTION	78
5.5	LE NIVEAU M ₂ : MODÉLISATION DU HMS	80
5.5.1	Représentation d'un HMS contrôlé par les ressources	81
5.5.2	Représentation d'un HMS contrôlé par le produit	81
5.5.3	Transformation pour la réingénierie	82
5.6	LIEN ENTRE LES TROIS DERNIERS NIVEAUX	82
5.6.1	MBCSA	82
5.6.2	ASCI	83
5.6.3	Bilan	85
5.7	MÉTHODOLOGIE DE RÉINGÉNIERIE DES HMS	85
5.7.1	Une méthodologie composée de deux parties	85
5.7.2	Utilisation à partir du méta-modèle	89
5.7.3	Utilisation à partir du MGC d'un domaine	90
5.8	CONCLUSION	92

5.1 INTRODUCTION

Ce chapitre présente notre approche méthodologique pour réorganiser un système de production que nous avons appelée *Méthodologie de Réingénierie des HMS (MRHMS)*. Comme nous l'avons signalé dans le chapitre 1, nous avons cherché les méthodologies déjà développées pouvant permettre la réingénierie des systèmes de production que nous souhaitons appliquer au domaine du circuit du médicament (chapitre 6). Dans les chapitres précédents, nous avons expliqué les concepts les plus récents sur l'organisation d'un système de production, que ce soit au niveau du flux de matière ou du flux d'information. Nous avons présenté les outils de modélisation les plus utilisés. Une partie de cette méthodologie appliquée au circuit du médicament a fait l'objet d'une publication [Huet 10b]. Dans ce chapitre, nous expliquerons nos choix et nous détaillerons l'approche proposée. Pour permettre le développement d'une méthodologie, nous devons définir une approche, une architecture et les langages utilisés. Ces points seront développés dans la seconde section. Puis, dans une troisième, nous reviendrons sur le vocabulaire employé pour éviter toute ambiguïté. Nous entrerons ensuite dans le détail de notre méthodologie dans les sections 4 à 6 en expliquant chaque étape avant de faire un récapitulatif dans la septième section.

5.2 CHOIX DES OUTILS DE MODÉLISATION

5.2.1 Choix de l'approche

Le terme « Développement à base de composants » est un terme populaire dans le domaine du génie logiciel [Barbier 02]. On a vu dans l'introduction du chapitre 3 que ce paradigme a aussi été utilisé dans le domaine des systèmes de production automatisés [Kisti 05]. L'approche par composant permet de prendre en compte les paramètres d'interopérabilité et de réutilisabilité.

La réutilisation est prise en compte car il s'agit de considérer les composants comme des briques de développement existantes, pouvant être réutilisées dans plusieurs situations. Les principes d'abstraction (vu boîte noire/boîte blanche) et de variabilité caractérisent l'approche orientée « réutilisation ». L'interopérabilité est essentielle pour permettre l'assemblage et la coopération de composants hétérogènes et souvent distribués.

Pour [Barbier 02], l'architecture est une spécification explicite des composants d'un système et de leurs interactions. La motivation pour définir une architecture est de fournir un plan précis (ou méta-modèle) approprié pour prédire le comportement d'un système avant de le construire et pour guider son développement.

Les composants sont définis comme des unités de composition qui décrivent et/ou assurent des fonctions spécifiques, possèdent des interfaces de besoins et des interfaces de services, et des contextes particuliers d'exécution. Ils peuvent être déployés indépendamment et composés avec d'autres composants. Ils sont préfabriqués, prétestés, s'auto-contiennent, et disposent de documentations appropriées et d'un statut de réutilisation bien défini. Les connecteurs représentent les interactions entre composants. Les connecteurs ont aussi des interfaces qui définissent les rôles joués par les divers participants à l'interaction. Les systèmes sont alors définis comme des configurations de composants et de connecteurs.

[Barbier 02] explique que les systèmes peuvent aussi être hiérarchiques : les composants et les connecteurs peuvent représenter des sous-systèmes qui possèdent des architectures internes. [Guzélian 05] présente aussi l'approche « patron » qui répond au besoin de formaliser et capitaliser des solutions à des problèmes récurrents de conception. Le patron est alors un

composant réutilisable. [Auzelle 09] utilise aussi une approche « patron » pour modéliser un SdS récursif et multi-échelle centré sur le produit. Il utilise le patron (pattern en anglais) « composite » présenté dans [Gamma 95] et représenté sur la figure 5.1.

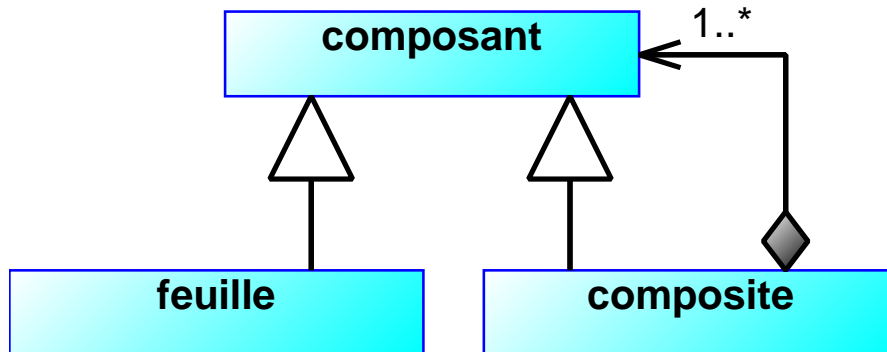


FIGURE 5.1 – Le « pattern composite » inspiré de [Gamma 95]

Par résumer, l’approche par composant présente l’avantage de gérer les problématiques d’interopérabilité et de réutilisabilité, ce qui correspond à nos problématiques. Nous nous axerons par conséquent sur cette approche.

5.2.2 Choix de l’architecture

La démarche MDA consiste en des transformations entre différents modèles qui représentent différents niveaux d’abstraction. Chaque transformation est un passage entre un Platform Independent Model (PIM) à un Platform Specific Model (PSM) et ceci jusqu’au code (cf section 3.2.4). Le modèle qui a le rôle de PSM lors d’une première transformation peut avoir le rôle d’un PIM lors d’une seconde [OMG 03]. Nous avons vu lors du chapitre précédent que dans plusieurs travaux proches de notre domaine, le nombre de niveaux d’abstraction utilisés est au nombre de 4. La démarche MDA utilisée dans ses travaux n’est pas la seule démarche à utiliser 4 niveaux d’abstraction. On peut citer par exemple la méthodologie Tropos développée pour la conception de SMA (cf section 3.3.4.2) qui propose une composition en 4 niveaux d’abstraction : Méta-Méta-Modèle, Méta-Modèle, Domaine, Instance. Le MDA propose aussi un patron (pattern) pour l’utilisation d’une transformation. Une transformation entre un PIM et un PSM est montré sur la figure 5.2.

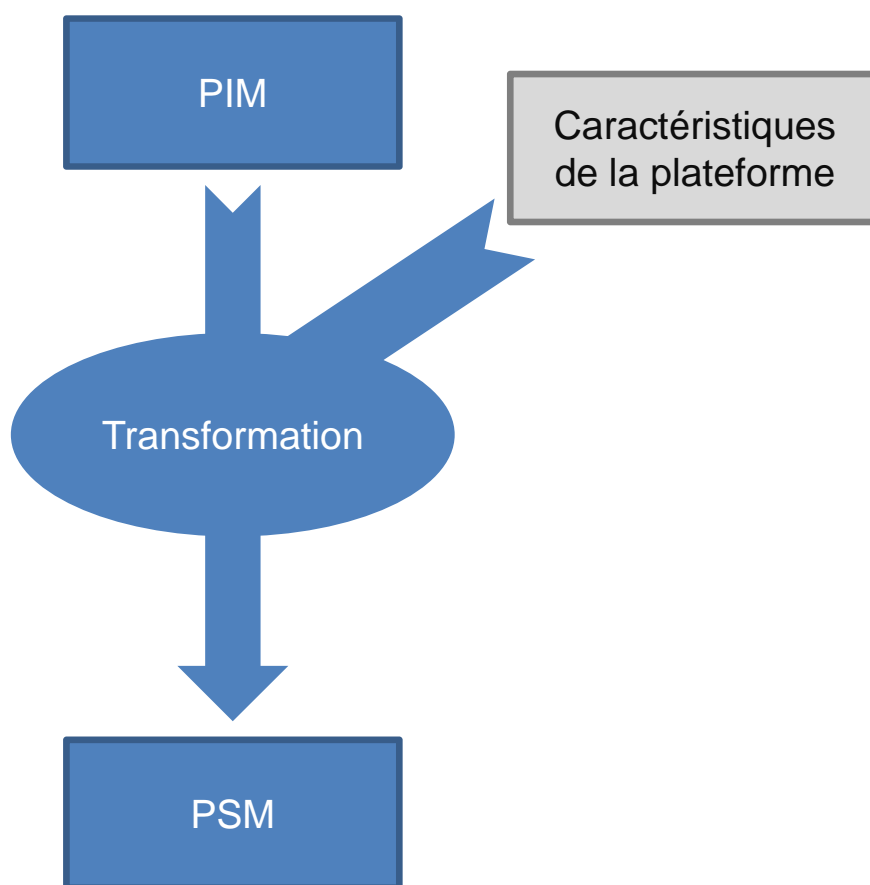


FIGURE 5.2 – Transformation de modèle

Pour permettre cette transformation, des éléments extérieurs (caractéristiques de la plateforme, autre modèle...) peuvent être intégrés (représenté par le cadre gris sur la figure 5.2).

Nous proposons de développer une méthodologie à partir de 4 niveaux d'abstraction pour permettre la réingénierie de HMS :

- nous définissons notre point de vue dans le méta-méta-modèle (M₃),
- dans le niveau M₂ (méta-modèle), nous montrons la structure possible pour différents HMS, c'est-à-dire pour une classe de HMS,
- dans le niveau M₁ (modèle d'un domaine), nous proposons d'instancier cette structure possible à un domaine en conformité avec une classe de HMS,
- dans le niveau M₀, nous proposons d'instancier le niveau M₁ à un système particulier du domaine.

Nous résumons notre démarche dans la table 5.1.

M ₃	Méta-méta-modèle	Description d'un système de production
M ₂	Méta-modèle	Une classe de HMS
M ₁	Modèle	Un domaine de la classe
M ₀	Univers du discours	Un système du domaine

TABLE 5.1 – Démarche MDA préconisée par notre méthodologie

5.2.3 Choix du langage

Nous recherchons un langage permettant une modélisation des composants de type holon et de pouvoir montrer leurs définitions et leurs relations dans le méta-méta-modèle. Ensuite, dans le méta-modèle, de pouvoir montrer leurs interactions dans différents contextes. Le langage qui nous paraît le plus adapté à ces contraintes est le langage **SysML**. Selon la documentation officielle [OMG 10a] disponible sur le site l'OMG¹, **SysML** supporte la spécification, l'analyse, la conception, la vérification d'un grand nombre de systèmes complexes. Ces systèmes peuvent inclure des systèmes physiques, des systèmes d'information, des processus, des personnes ou des biens. L'origine de **SysML** est due à International Council on Systems Engineering's (INCOSE) par le groupe de travail « Model Driven Systems Design » créé en janvier 2001 pour personnaliser le langage **UML** pour une application à l'ingénierie-système (figure 5.3).

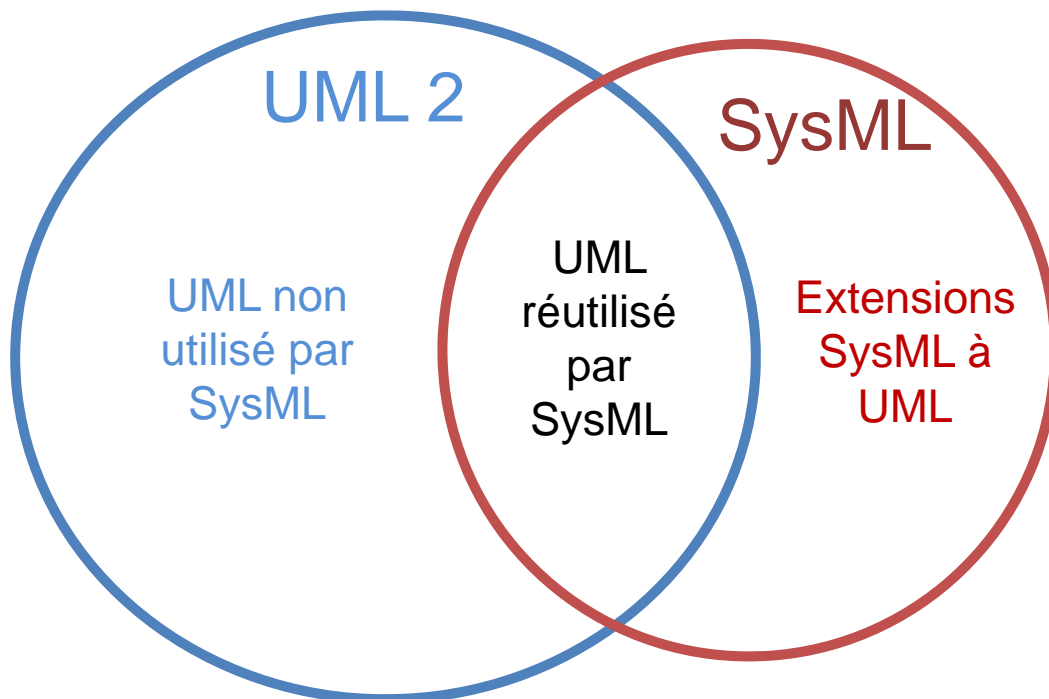


FIGURE 5.3 – Vue d'ensemble de la relation entre UML et SysML inspirée par [OMG 10a]

Les blocs **SysML** peuvent être utilisés à toutes les étapes de spécification et de conception d'un système, et peuvent être appliqués à beaucoup de types de systèmes. Cela inclut la modélisation des décompositions logiques ou physiques de systèmes, et la spécification d'éléments logiciels, physiques ou humains. Des parties de ces systèmes peuvent interagir de différentes façons, telles que des opérations logiciels, des transitions à événement discret, des flux d'entrée/sortie, ou des interactions continues. Le *Block Definition Diagram - diagramme de définition de bloc* (**BDD**) de **SysML** définit les caractéristiques des blocs et les relations entre blocs telles que les associations, les généralisations et les dépendances. Il représente la définition des blocs en terme de propriétés et d'opérations, et les relations avec des systèmes

1. <http://www.omg.sysml.org> (accessible au 29 juillet 2010)

telles que les relations hiérarchiques ou les arbres de classification. Le **BDD** est basé sur le diagramme de classe d'**UML**, avec des restrictions et des extensions.

Le *Internal Block Diagram - diagramme interne de bloc* (**IBD**) de **SysML** représente les structures internes des blocs en terme de propriétés et de connections entre propriétés. Un bloc peut inclure des propriétés pour spécifier ces valeurs, ces parties et des références vers d'autres blocs. Le **IBD** est basé sur le diagramme de structure composite d'**UML** avec des restrictions et des extensions.

[Chiron 09] utilise **SysML** pour permettre la conception de composants multi-facettes. Chaque facette peut être représentée par différents diagrammes **SysML** : diagramme d'activité, état/transion pour la partie dynamique et diagramme **IBD** pour la partie statique. Ceci lui permet de décomposer chaque élément en plusieurs nouveaux composants et de répéter cette étape plusieurs fois si besoin, suivant la complexité de l'élément étudié.

[Naumenko 03] explique que le méta-méta-modèle doit être conçu pour permettre la définition de l'existant qui a de l'intérêt lors de la modélisation des méta-modèles, et permettre leur unification sur un cadre de travail commun. Un diagramme tel que le **BDD** nous apparaît parfaitement adapté pour effectuer cette définition. Logiquement, les différents scénarios correspondant aux méta-modèles seront mis en place par les **IBD** qui permettent d'explicitement facilement une organisation des éléments définis dans le **BDD**. Pour les niveaux d'abstraction inférieurs, nous effectuerons une instanciation des **IBD** de niveau supérieur.

5.3 VOCABULAIRE EMPLOYÉ

5.3.1 Mise au point sur le terme « générique »

La terme générique peut avoir plusieurs sens, c'est pourquoi il nous semble important de le définir clairement. Si nous prenons le sens générique de la méthodologie **ASCI**, un modèle de connaissance générique représente un ensemble d'entités applicable à plusieurs systèmes (réutilisables).

Dans la littérature sur la modélisation d'entreprise, le terme générique a pris un autre sens, il correspond au méta-modèle d'un système dans le sens où il correspond au niveau où tous les éléments sont définis et décrits [Roux 10a]. On spécialise après ce méta-modèle pour un champ d'application particulier. Dans [Ulmer 10], un compromis nous semble intéressant : c'est la notion de « généricité relative » et de « généricité absolue ». La « généricité absolue » correspond au niveau de la description (généralement le méta-modèle) et lors de la spécialisation, on utilise le terme de « généricité relative » pour un champ d'application particulier. La généricité relative revient donc à la même définition que la généricité dans **ASCI**, c'est-à-dire qu'un modèle de connaissance générique d'un domaine est un modèle de connaissance réutilisable pour tous les systèmes de ce domaine. Quand nous emploierons le terme « générique » dans cette thèse, il correspondra au sens **ASCI** et donc au sens de « généricité relative » pour la modélisation d'entreprise tel que définie par [Ulmer 10].

5.3.2 Transformation entre niveaux d'abstraction

De nombreux termes sont utilisés pour faire le lien entre les quatre niveaux d'abstraction : « spécialisation, instanciation et implantation » pour [Ulmer 10], « instanciation, particularisation et évaluation » pour [El Haouzi 08a]. D'autres auteurs proposent aussi le terme « instanciation » (instantiate) à chaque transformation [Mekki 10]. L'avantage d'utiliser un terme

différent à chaque transformation est de permettre une différenciation claire de chacune des étapes.

Généralement, le terme « instanciation » correspond au passage d'un domaine (classe de systèmes) à un système particulier. Cela correspond bien à ce que nous voulons effectuer en passant du méta-modèle (M2) au modèle (M1), puis en passant du modèle au système (Mo). Par contre, la transformation du méta-méta-modèle (M3) au méta-modèle (M2) n'est pas une simple instanciation car il correspond à une étude des différentes possibilités de configuration à partir de définitions et de descriptions. Nous considérons cette étape plus comme une « spécialisation ». Nous spécialisons ces définitions pour un champ d'application donné.

5.4 LE NIVEAU M3 : DESCRIPTION D'UN SYSTÈME HOLONIQUE DE PRODUCTION

Nous avons représenté au niveau M3 les concepts et les relations pour un système holonique de production. Nous avons cherché à avoir une généricité relative aux systèmes de production. Nous utilisons à ce niveau un BDD qui nous semble le diagramme le plus adapté pour effectuer ce type de description. Il est montré sur la figure 5.4.

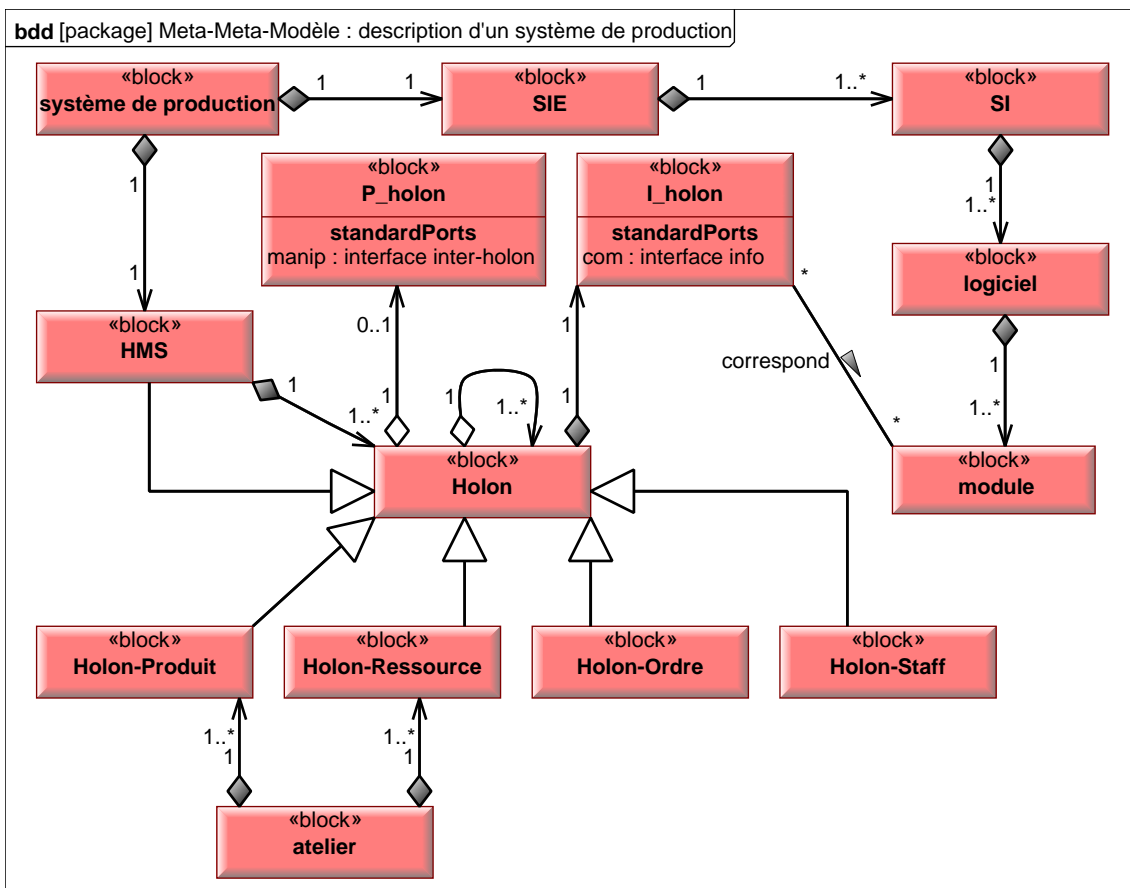


FIGURE 5.4 – Le Méta-Méta-Modèle décrivant les relations de base dans un système holonique de production

A partir des travaux présentés dans les chapitres 2 et 4, nous décrivons les relations de base entre les quatre types d'holons. Un holon peut être de quatre types : *Holon-Ordre* (HO), *Holon-Ressource* (HR), *Holon-Produit* (HP) et *Holon-Staff* (HS). Un HMS est composé de plusieurs holons et en est un lui-même : nous avons alors utilisé le patron défini dans [Gamma 95] plus haut. Un holon peut être lui-même décomposé en plusieurs sous-holons. Un holon est composé d'une partie physique que nous avons appelée P_holon et d'une partie informationnelle que nous avons appelée I_holon. Un I_holon correspond à un module d'un logiciel du SIE. Un holon peut ne pas avoir de partie physique mais a obligatoirement au moins une partie informationnelle. Nous avons remarqué qu'un SIE est divisé en plusieurs SI (chaque service d'une entreprise a son propre SI). Un logiciel peut être utilisé par plusieurs SI. Une entreprise est composée d'un HMS.

SysML nous permet d'avoir la notion de port. Chaque partie physique d'un holon a un port d'entrée/sortie correspondant à une manipulation manuelle ou automatique. Cette manipulation physique peut être un service rendu à un autre holon (un robot qui assemble une pièce par exemple) ou une modification de la partie physique (opération d'assemblage ou de transformation). Ce port correspondra donc à une relation inter-holons. La partie informationnelle d'un holon (généralement : un module d'un logiciel) a un flux informationnel d'entrée/sortie qui permettra de gérer le flux d'information du holon. Un holon pouvant être composé d'autres holons, ce flux informationnel pourra aussi correspondre au flux intra-holon autant qu'inter-holons. Un module peut faire partie d'un HMS et/ou d'un SIE. Cela dépend de la vue que l'on souhaite adopter. Un informaticien verra le SI d'une ressource comme un logiciel comme un autre, tandis que si on adopte une vision holonique, un module de ce logiciel (le module qui pilotera la ressource) fera partie d'un holon. Certains modules, les parties informationnelles d'un holon, sont donc à la fois une composante du SIE et à la fois une composante du HMS et de l'atelier. L'atelier de l'entreprise est composé des HP et des HR. Les I_holon correspondront au module de logiciel utilisé par les HR et les HO. Par conséquent, la représentation d'un HMS sera composée d'éléments appartenant à l'atelier et au SIE.

L'atelier sera composé des holons qui ont une partie physique et qui permettent de représenter un HMS avec le SIE qui sera composé des holons ayant seulement une partie informationnelle.

5.5 LE NIVEAU M2 : MODÉLISATION DU HMS

Nous faisons le lien entre le niveau M3 et le niveau M2 par une transformation telle que décrit sur la figure 5.5.

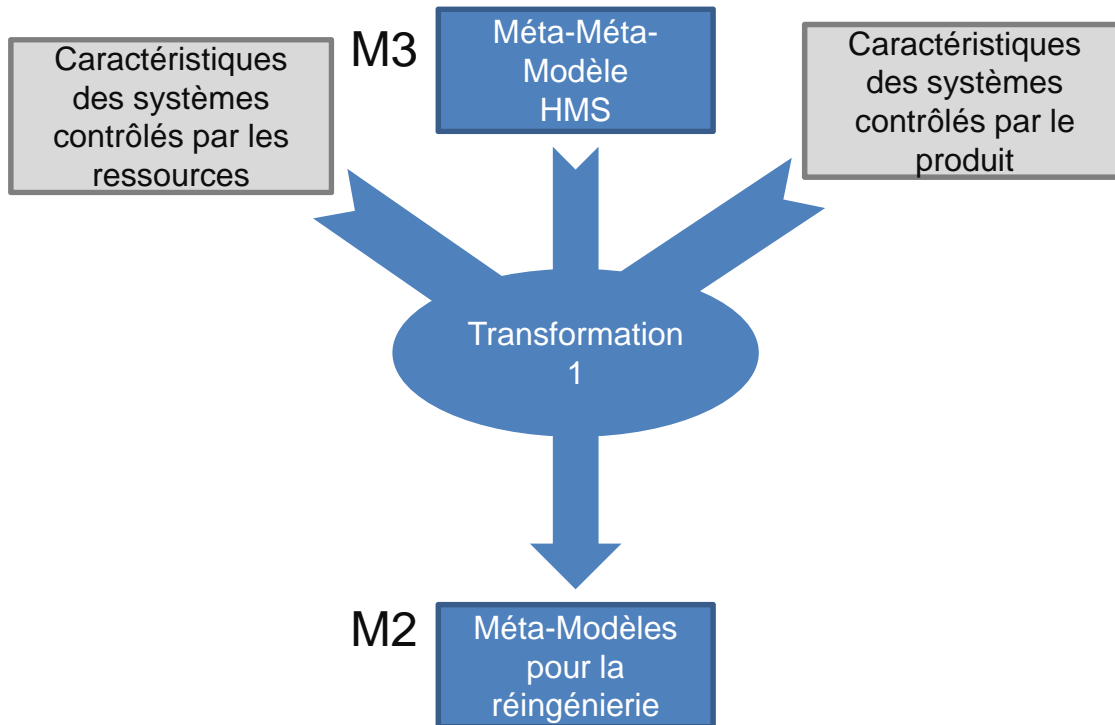


FIGURE 5.5 – Transformation entre le niveau M3 et M2

Ces méta-modèles représentent un point de vue HMS du niveau M3. D'autres points de vue auraient pu être mis en avant comme par exemple la représentation du SIE. Cela aura donné lieu à des méta-modèles différents.

Plusieurs fonctionnements de HMS sont possibles à partir des relations entre les holons et des descriptions représentées en M3. Nous souhaitons effectuer une réingénierie à partir d'un fonctionnement contrôlé sur les ressources pour aller vers un fonctionnement contrôlé par le produit. Le contrôle par le produit permettant une meilleure flexibilité, réactivité et traçabilité du système. Par conséquent, nous proposons deux méta-modèles :

- **MM1** Représentant les relations entre holons dans un atelier contrôlé par les ressources,
- **MM2** Représentant les relations entre holons dans un atelier contrôlé par les produits.

Nous choisissons les IBD de SysML car ils permettent à partir d'un BDD de décrire une structure correspondant à un contexte précis. Ces méta-modèles sont écrits à partir de la description faite dans le chapitre précédent d'un système contrôlé par les ressources, et d'un système contrôlé par le produit.

5.5.1 Représentation d'un HMS contrôlé par les ressources

Nous avons cherché à avoir ici une généralité relative à un HMS contrôlé par les ressources. Nous montrons figure 5.6 le modèle que nous proposons.

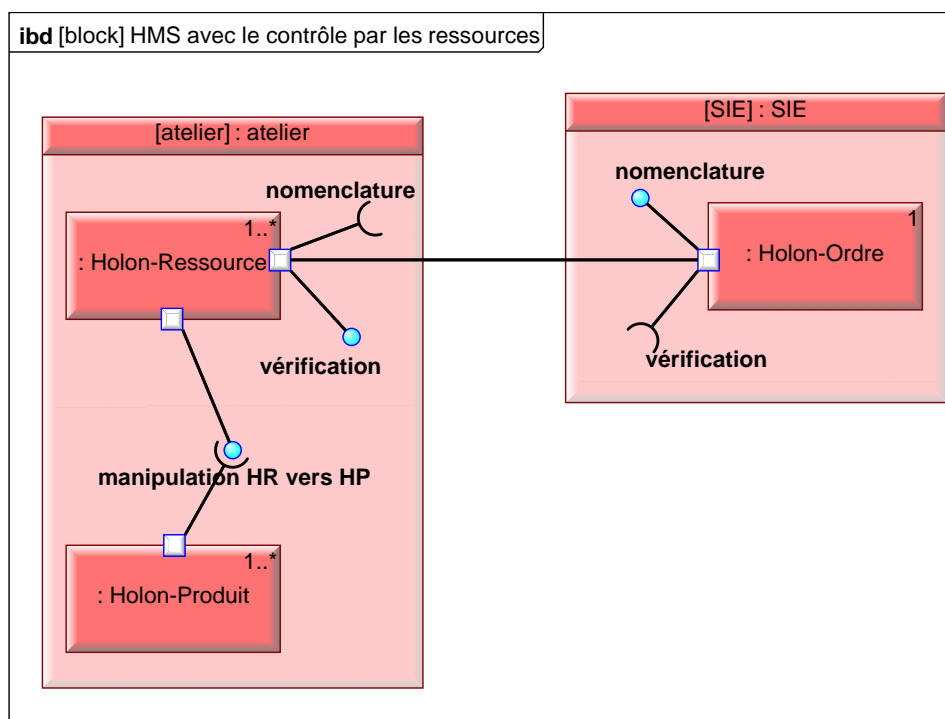


FIGURE 5.6 – Méta-Modèle MM₁, montrant les relations dans un système de production contrôlé par les ressources

Nous proposons ici de représenter une vue d'un système de production à partir d'un HMS, c'est-à-dire que nous représentons seulement les holons qui composent l'entreprise. Nous nous sommes inspiré du travail de [Leitão 04] pour étudier ce type de fonctionnement. Pour le HO, il n'a qu'une partie informationnelle, c'est le module du logiciel du SI qui envoie la planification à l'atelier et qui permet la vérification lors du contrôle du produit. Dans le cas du contrôle par les ressources, cette donnée est réceptionnée par les HR de l'atelier. Les ressources dirigent alors la production du HP.

5.5.2 Représentation d'un HMS contrôlé par le produit

Nous avons cherché à avoir ici une généralité relative à un HMS contrôlé par le produit. Le modèle que nous proposons est montré sur figure 5.7.

Nous nous sommes inspiré du travail présenté dans la section 4.3 pour étudier ce type de fonctionnement. Le HP reçoit l'ordre de fabrication de l'HO qui le crée. Lors d'une phase de contrôle du produit, celui-ci demande à l'HO une vérification pour voir si des changements ou des modifications d'assemblage ou de transformation ont été effectués. Le HP a une relation de dépendance avec certains HR et une relation de réalisation avec certains autres. Nous voyons une interface fournie comme une interface qui « agit » sur un autre holon (qui amène des transformations ou des modifications). Par exemple, quand le HP reçoit des éléments lors d'une transformation ou d'un assemblage. Le HP fournit une interface à d'autres HR ;

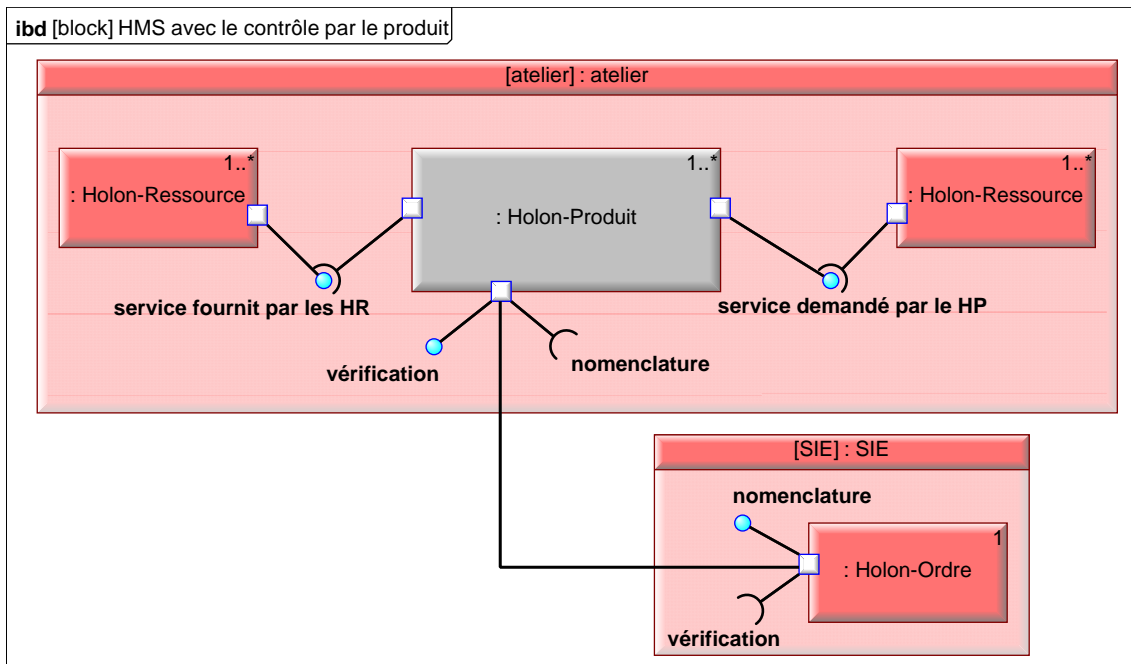


FIGURE 5.7 – Méta-Modèle MM2, montrant les relations dans un système de production contrôlé par le produit

dans le cas où une ressource est un système de manutention, il demande à la ressource de le transporter. Dans ce cas, ces HR attendent qu'un HP les sollicite et ces HR ne transforment pas le HP.

5.5.3 Transformation pour la réingénierie

La documentation officielle du MDA [OMG 03] propose aussi une transformation entre modèles du même niveau. Il y a par conséquent deux types de transformation, une transformation « verticale », pour passer d'un niveau d'abstraction à un autre, et une transformation « horizontale » qui explicite les règles pour passer d'un modèle à un autre à un même niveau d'abstraction. Par exemple, [Baïna 06a] propose de faire le lien entre son méta-modèle décrivant un holon et celui du langage UEML. On peut citer aussi [Ulmer 10], qui propose une transformation à un même niveau d'abstraction pour passer d'un langage de modélisation d'entreprise à un autre (BPA et BPI). Le processus de réingénierie correspond à cette transformation d'un méta-modèle en un autre. **Notre méthodologie permet d'identifier les composants d'un système contrôlé par les ressources, puis de l'adapter au méta-modèle représentant un SCP pour pouvoir le réorganiser.**

5.6 LIEN ENTRE LES TROIS DERNIERS NIVEAUX

5.6.1 MBCSA

Nous avons recherché les méthodologies qui pourraient nous aider à concevoir un *Modèle Générique de Connaissance* (MGC) d'un domaine et de l'instancier à un système de ce domaine dans le but de faire le lien entre le niveau M2 et M1 puis entre M1 et Mo. Deux des méthodologies présentées au chapitre 3 nous semblent intéressantes : ASCI et MBCSA.

La *Méthodologie Bidirectionnelle de Conception des Systèmes Automatisés* (MBCSA) (cf section 3.3.3) propose une division des systèmes automatisés en composants multi-facettes [Chiron 09] :

- Facette de Dialogue Homme-Machine,
- Facette logique de fonctionnement,
- Facette de description physique ou mécanique,
- Facette de description électrique,
- Facette MES.

Cette structure de composants ayant des facettes informationnelles est proche du concept d'holon. Nous pouvons faire la correspondance entre un composant multi-facettes et un holon tel que décrit par [Mcfarlane 00] dans la figure 2.3 de la section 2.3 (table 5.2).

Facette de composant	Facette d'un holon
Dialogue Homme-Machine	Interface utilisateur
Logique de fonctionnement	Centre de décision
Description physique ou mécanique	Contrôle physique
Description électrique	Processus physique
MES	Interface inter-holon

TABLE 5.2 – Comparaison des facettes de composants vues par [Chiron 09] avec un holon vu par [Mcfarlane 00]

Il propose aussi que ces facettes soient réutilisables en étant adaptables à plusieurs plateformes cibles, comme le préconise la démarche MDA. Cette méthodologie propose deux formes d'analyse, une montante et une descendante (figure 3.4 dans la section 3.3.3). L'approche montante passe par les phases de :

- Identification des briques élémentaires : faire la liste de tous les éléments d'une entreprise,
- Spécification multi-facettes des briques élémentaires : chaque composant est décrit sous les différentes facettes qui sont ensuite « éclatées » pour retrouver les composants encapsulés dans les différentes facettes,
- Validation : test de chaque composant sur une plate-forme,
- Capitalisation : Une fois les différentes facettes spécifiées et le comportement de l'élément validé, ce dernier peut être stocké dans la bibliothèque d'éléments disponibles.

La bibliothèque de composants est ainsi construite. L'approche descendante utilise cette bibliothèque en suivant les phases de :

- Étude des besoins : analyse classique d'identification des besoins auxquels le système devra répondre,
- Réutilisation : identification des composants dont le système aura besoin,
- Conception : assemblage des composants entre eux,
- Déploiement : mise en place du système.

5.6.2 ASCI

La démarche méthodologique *Analyse, Spécification, Conception, Implémentation* (ASCI) [Gourgand 91] est issue des travaux de modélisation du LIMOS sur les systèmes complexes

(système de trafic urbain, système de production de biens et services, systèmes hospitaliers, ...). Elle permet de concevoir une méthodologie de modélisation d'un domaine (définition 5.1), le modèle générique de connaissance de ce domaine, et de réaliser une bibliothèque de composants logiciels qui est exploitée pour générer un modèle d'action pour un système du domaine.

Définition 5.1 *Un domaine est une classe de systèmes qui regroupe un ensemble de systèmes comparables, ayant les mêmes caractéristiques techniques et fonctionnelles. Il est ainsi possible de décrire la classe des systèmes industriels, la classe des systèmes hospitaliers, ... [Rodier 10]*

Cette démarche a été construite à la suite de nombreuses études concernant la modélisation et l'évaluation des systèmes complexes (exemples : systèmes industriels, systèmes de transport, ...).

Le processus de modélisation d'un système du domaine préconise la construction consécutive d'un modèle de connaissance (modèle de description du fonctionnement du système) et d'un ou plusieurs modèles d'action (modèles informatiques) qui sont obtenus à partir du modèle de connaissance. La construction puis l'utilisation des modèles de connaissance et d'action pour la conception, la configuration, le pilotage d'un système complexe constituent le processus de modélisation de ce dernier.

Pour la construction des modèles de connaissance des systèmes complexes, **ASCI** préconise une décomposition systémique inspirée par [Le Moigne 92] du système étudié en trois sous-systèmes communicants :

- le *Sous-Système Physique (SSP)* définit l'ensemble des moyens physiques (pour la production, le stockage, la manutention et le transport), leur répartition géographique et leur interconnexion,
- le *Sous-Système Logique (SSL)* est constitué des entités que le système doit traiter, des ensembles d'opérations concernant ces flux, ainsi que des nomenclatures qui s'y rapportent,
- le *Sous-Système Décisionnel (SSD)* contient les règles de gestion et de fonctionnement du système.

Cette méthodologie, particulièrement adaptée à la modélisation des systèmes complexes, a été largement utilisée. On peut citer par exemple :

- pour les systèmes hospitaliers, les travaux de Catherine Combes [Combes 94] et Christine Di Martinelly [Di Martinelly 08]. Elle a aussi été utilisée dans le cadre du projet de modélisation du nouvel hôpital d'Estaing à Clermont-Ferrand avec Pierre Féliès [Féliès 06], Julie Chauvet [Chauvet 09] et Sophie Rodier [Rodier 10],
- pour les systèmes multi-agents, on peut citer les travaux de Jean-Yves Goujon [Goujon 97], Philippe Lacomme [Lacomme 98] et Ahmad Ali [Ali 09],
- pour les systèmes de transport et de trafic urbain, ceux de David Sarramia [Sarramia 02], Jean-Marie Boussier [Boussier 07] et Patrice Leclaire [Leclaire 09].
- pour les systèmes complexes avec un pilotage hybride (avec une partie décentralisée et une partie centralisée), ceux de Hind El Haouzi [El Haouzi 08a], où il est proposé d'enrichir la décomposition systémique par un modèle représentant les *Entités Décisionnelles (ED)* pour faire le lien entre le **SSP** et le **SSD** (cf section 4.3).

5.6.3 Bilan

Nous avons vu que les composants développés par la méthodologie **MBCSA** sont proches des holons, mais cette méthodologie a été appliquée seulement à un système de production composé d'automates dont le côté « informationnel » et les relations d'interopérabilité avec le **SI** ont peu été développés.

De son côté, la méthodologie **ASCI** a donc déjà fait ces preuves sur des **SMA**, sur le milieu hospitalier et sur les systèmes à pilotage hybride. Elle nous semble par conséquent parfaitement adaptée pour la modélisation de systèmes holoniques. Comme inconvénient, on peut signaler qu'elle n'a pas été éprouvée pour des systèmes automatisés et demande quelques adaptations pour être parfaitement opérationnelle sur un système qui est composé d'entités ayant des éléments physiques et informationnels.

Dans [Huet 09a], nous avons proposé une nouvelle méthodologie à partir de la méthodologie **ASCI** et de l'approche par composants, en se basant sur le constat que les méthodologies **MBCSA** et **ASCI** étaient proches et se complétaient. Nous détaillons cette méthodologie dans la prochaine section.

5.7 MÉTHODOLOGIE DE RÉINGÉNIERIE DES HMS

5.7.1 Une méthodologie composée de deux parties

Un **HMS** a une partie correspondant à un système automatisé et une partie logicielle. Nous cherchons à pouvoir instancier un **HMS** à un domaine puis à un système de ce domaine.

La méthodologie **MBCSA** utilise l'approche par composant pour la conception de composants multi-facettes pour les systèmes automatisés. **ASCI** utilise la notion de composant lors de la construction d'une bibliothèque de composants logiciels réutilisables. **MBCSA** utilise plus en profondeur la notion de composant, avec une décomposition de chaque facette de composant en vue « boîte noire » puis « boîte blanche », c'est-à-dire un « accès » au niveau d'abstraction inférieur en représentant d'autres composants sous-jacents lors de l'étape de Spécification. L'approche montante (Analyse Objet Montante (AOM)) préconisée par **MBCSA**, qui va jusqu'à la capitalisation des composants dans une bibliothèque, est identique aux étapes d'Analyse, de Spécification, de Conception et d'Implémentation qui permettent la construction d'une bibliothèque de composants logiciels dans **ASCI** pour un domaine (classe de systèmes). Nous avons comparé ces deux approches sur la figure 5.8 et dans la table 5.3.

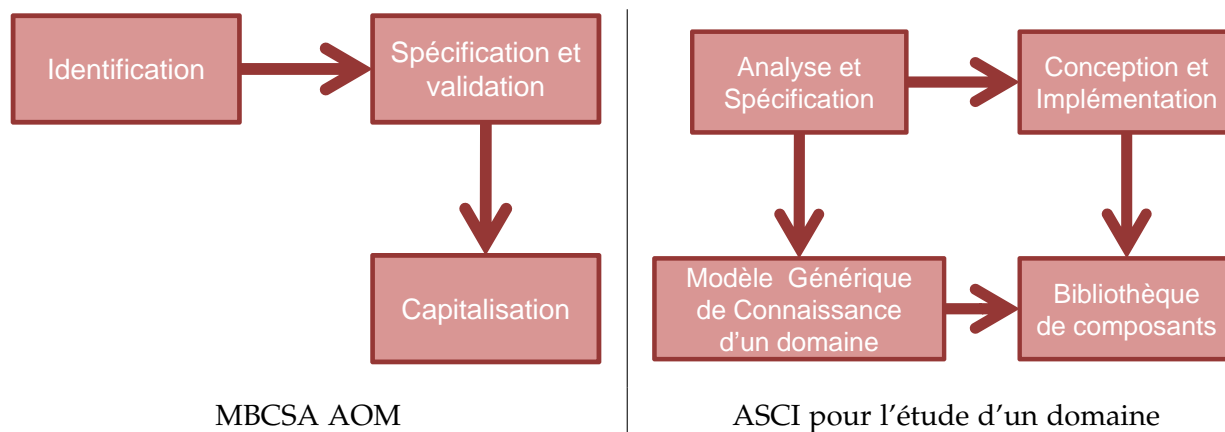


FIGURE 5.8 – Comparaison entre l'approche montante de MBCSA et l'étude d'un domaine avec ASCI

MBCSA AOM	ASCI
Identification	Analyse et Spécification
Spécification	Conception
Validation	Implémentation
Capitalisation	Bibliothèque de composants

TABLE 5.3 – Comparaison de l’approche montante de MBCSA et d’ASCI

L’approche descendante (Analyse Objet Descendante (AOD)), qui réutilise les composants de cette bibliothèque jusqu’au déploiement, est identique aussi à l’élaboration d’un modèle d’action d’un système du domaine dans **ASCI**, qui réutilise les éléments de la bibliothèque de composants logiciels. Nous avons représenté ces différences sur la figure 5.9 et dans la table 5.4.

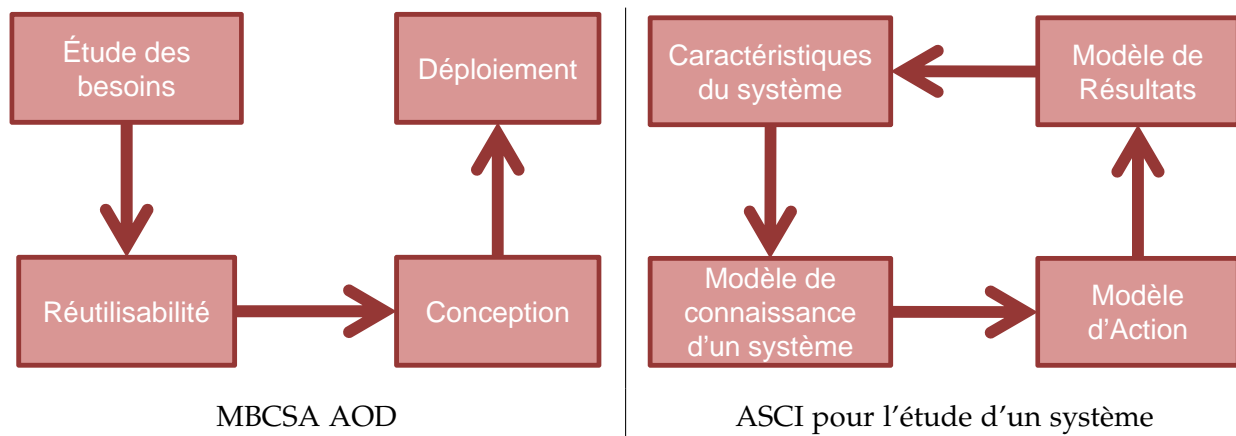


FIGURE 5.9 – Comparaison entre l’approche descendante de MBCSA et l’étude d’un système avec ASCII

MBCSA AOD	ASCI
Étude des besoins	Caractéristique du système
Réutilisation	Modèle de Connaissance
Conception	Modèle de Connaissance
Déploiement	Modèle d’action et de résultats

TABLE 5.4 – Comparaison de l’approche descendante de MBCSA et d’ASCI

De plus, la proposition faite dans [El Haouzi o8b] de compléter la décomposition systématique classique proposée dans **ASCI** par un diagramme représentant les *Entités Décisionnelles (ED)* nous paraît une bonne solution pour représenter l’instanciation des méta-modèles de niveau supérieur.

ASCI est plus complet (boucle dans le processus de modélisation et élaboration d’un modèle de connaissance générique pour un domaine) car cette méthodologie a été conçue pour être adaptable à un maximum de domaines, contrairement à **MBCSA** qui a été conçu seulement pour la conception de systèmes automatisés. L’approche multi-facettes utilisée avec des

diagrammes de composant UML et des BDD et IBD de SysML nous apparaît, par contre, adaptée pour la conception de système holonique. Nous proposons d'utiliser les langages et la démarche préconisés par MBCSA pour la construction des diagrammes de ED.

Nous proposons, sur ces constats, de généraliser MBCSA et donc de compléter ASCI pour qu'il puisse être compatible avec l'approche développée dans MBCSA.

Par conséquent, nous avons développé une nouvelle méthodologie basée sur ces deux méthodologies, dans le but de permettre de résoudre le problème de cette double instanciation (M2 à M1 puis M1 à Mo) : *ASCI pour la Réingénierie des HMS* (ASCI-RHMS). Nous l'avons représenté sur la figure 5.10 en mettant en rouge et dans une police différente les changements par rapport au ASCI traditionnel présenté dans la section 3.3.2.

Nous avons structuré les principales phases de la façon suivante :

- La phase d'Analyse et de Spécification devient les vues en boîtes noires,
- La phase de Conception et d'Implémentation devient les vues en boîtes blanches de chaque composant.

La plus importante amélioration que nous proposons est d'améliorer la décomposition systémique préconisée par ASCI par un diagramme représentant les ED, comme proposé dans [El Haouzi 08b].

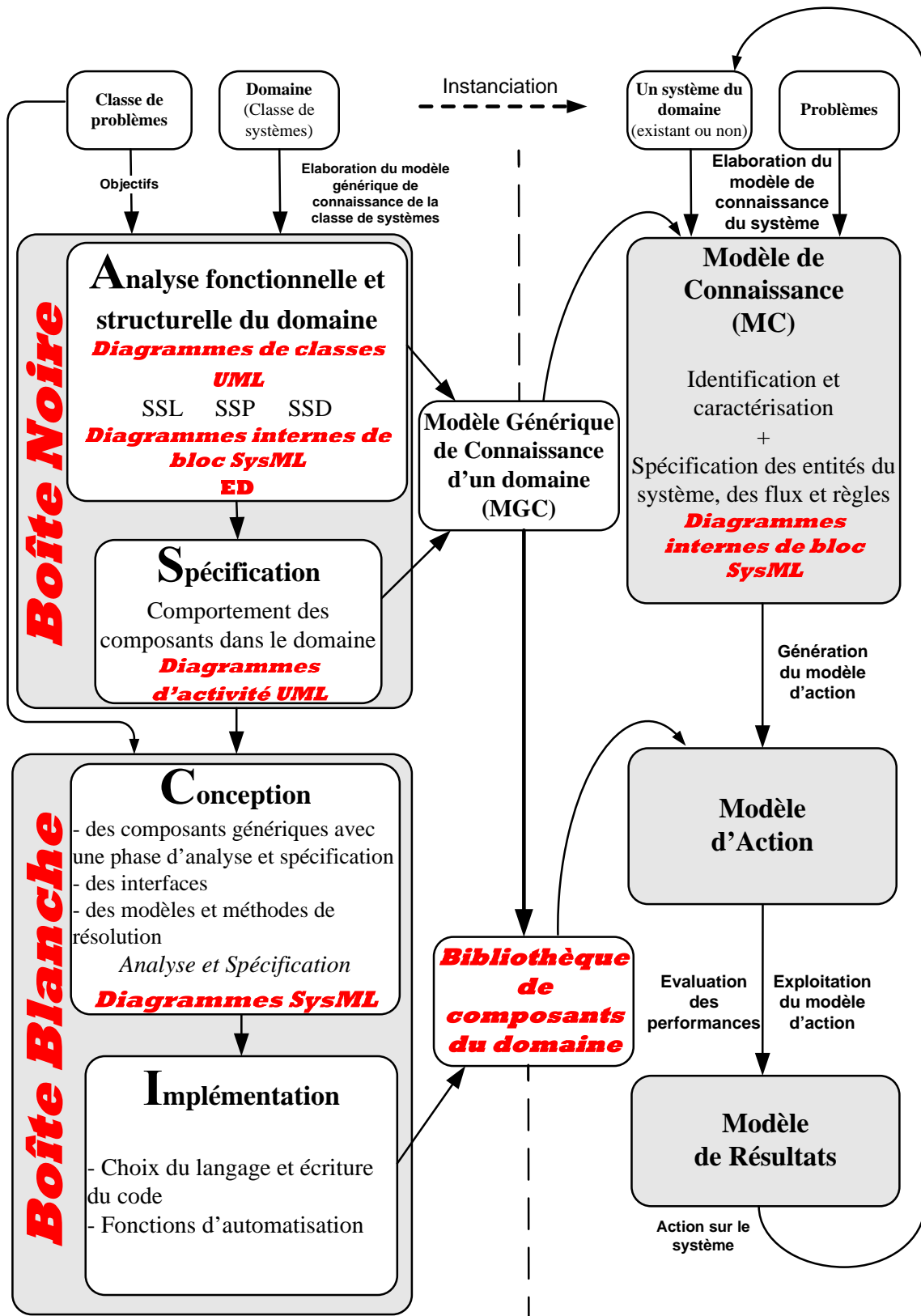
[Chiron 09] a signalé que dans le cas d'une interaction plus complexe entre deux composants, il est nécessaire de définir un nouvel objet intermédiaire. Les étapes de Conception et d'Implémentation de notre méthodologie consisteront en de nouvelles étapes d'Analyse et de Spécification pour un niveau plus fin de détail si besoin. Les « composants sur étagères » (Commercial off-the-Shelf - COTS), c'est-à-dire les composants propriétaires achetés à une entreprise dont on ne connaît pas les fonctionnements internes, ne pourront être décomposés et resteront en boîte noire. Les composants qui auront comme objectif d'être développés lors de différentes études devront être détaillés. La méthodologie MaSE présentée lors à la section 3.3.4.1 est un bon exemple de démarche à suivre lors d'une nouvelle phase d'Analyse et de Spécification pour chaque holon. Elle met en place une démarche à l'aide de diagramme UML pour la conception d'agent.

La méthodologie ASCI-RHMS pourra modéliser, simuler et déployer de façon claire et cohérente n'importe quel système qui peut se décomposer lui-même en composants, comme un système holonique. Nous proposons dans cette méthodologie une analyse systémique classique qui nous permettra d'étudier une classe de systèmes et d'isoler les ED. Ainsi, nous pourrions concevoir les modèles qui correspondront au niveau M1 (*Modèle Générique de Connaissance* (MGC) d'un domaine) en conformité avec les méta-modèles décrits dans ce chapitre. Les modèles qui seront en conformité avec les méta-modèles correspondront aux diagrammes des ED. Ce diagramme sera par conséquent modélisé avec un IBD où les composants anonymes des diagrammes 5.7 et 5.6 seront instanciés pour un domaine.

Si nous résumons notre méthodologie finale, que nous avons appelée *Méthodologie de Réingénierie des HMS* (MRHMS), on peut la décomposer en deux « sous-méthodologies » distinctes :

- La méthodologie que nous appellerons *MDA pour la Réingénierie des HMS* (MDA-RHMS), basée sur l'architecture MDA et une division en 4 niveaux d'abstraction,
- La méthodologie ASCI-RHMS, inspirée par ASCI et MBCSA, qui va permettre les transformations dans MDA-RHMS et la construction d'un modèle d'action (simulation, mise en œuvre...).

ASCI-RHMS permet d'effectuer les transformations entre les méta-modèles (niveau M2) et le MGC, et ainsi d'isoler les composants dont le fonctionnement devra être détaillé pour permettre un contrôle par le produit. Le diagramme des ED qui découle de l'instanciation des



Légende : **Apport de ASCI-RHMS par rapport à ASCI**

FIGURE 5.10 – Méthodologie ASCI-RHMS [Huet 10c]

méta-modèles et de l'élaboration de la **MGC** d'un domaine permettront de voir les relations et les interactions qu'il faut avoir entre les différents composants du système. Les diagrammes des **ED** permettront aussi de définir le niveau d'interopérabilité souhaité entre les différents **SI**.

D'autres méthodologies pourraient aussi être développées à partir de la vision d'un système de production décrit au méta-méta-modèle du niveau M₃, pour faire par exemple autre chose que de la réingénierie ou d'autres types de réingénierie (réingénierie du **SIE** par exemple). Il faudrait alors définir d'autres méta-modèles. Notre objectif étant la réingénierie pour le contrôle par le produit, nous nous intéressons dans le reste de ce manuscrit seulement à la méthodologie **MRHMS**. Nous nous concentrerons dorénavant à notre méthodologie à partir du niveau M₂ (méta-modèle). Il y a trois possibilités d'utilisation de **MRHMS** :

1. à partir du méta-modèle pour l'élaboration d'un *Modèle Générique de Connaissance* (**MGC**) d'un domaine (transformation 2),
2. à partir du **MGC** pour l'élaboration d'un *Modèle de Connaissance* (**MC**) d'un système appartenant au domaine (transformation 3) dans le cas où le **MGC** correspondant a déjà été réalisé,
3. en faisant une étude complète d'un système et de son domaine, c'est-à-dire en utilisant les deux possibilités précédentes.

Nous allons détailler ces possibilités dans les sections suivantes.

5.7.2 Utilisation à partir du méta-modèle

Une fois le niveau de modélisation (correspondant à la classe de problèmes à résoudre) et le domaine définis, la méthodologie **ASCI** consiste à concevoir un **MGC** d'un domaine (figure 5.11).

Le **MGC** d'un domaine préconisé par la méthodologie **ASCI** et repris dans **ASCI-RHMS** résulte de deux étapes successives [Rodier 10] :

- L'Analyse fonctionnelle et structurelle du domaine qui formalise le domaine sous une forme graphique ou syntaxique. Elle consiste à rechercher les entités du domaine, leurs fonctions et leurs associations. Cette analyse donne une vue « statique » du domaine étudié à partir de la décomposition systémique du système en trois sous-systèmes,
- La Spécification des entités et de leurs comportements, qui permet l'expression du fonctionnement des entités du domaine et des flux. En particulier, le système de pilotage doit être spécifié de façon précise, compte tenu des éléments à prendre en compte (règles de gestion. . .). La spécification ainsi réalisée doit permettre aux experts en modélisation et aux experts du système de s'accorder sur le fonctionnement (réel ou désiré) du domaine.

Nous proposons de rajouter une étape de transformation (transformation 2) qui correspond à l'isolation des éléments de la décomposition systémique correspondant aux méta-modèles présentés à la section 5.5. Ils permettront la formalisation des diagrammes représentant les **ED**. Nous avons alors entièrement réalisé le **MGC** d'un domaine qui correspond au niveau M₁ de notre méthodologie générale. Les diagrammes représentant les **ED** seront par conséquent des instances des méta-modèles, c'est-à-dire que :

- Un diagramme représentera le domaine contrôlé par les ressources avec les instances d'holons correspondant,
- Un diagramme représentera le domaine contrôlé par le produit avec les instances d'holons correspondant.

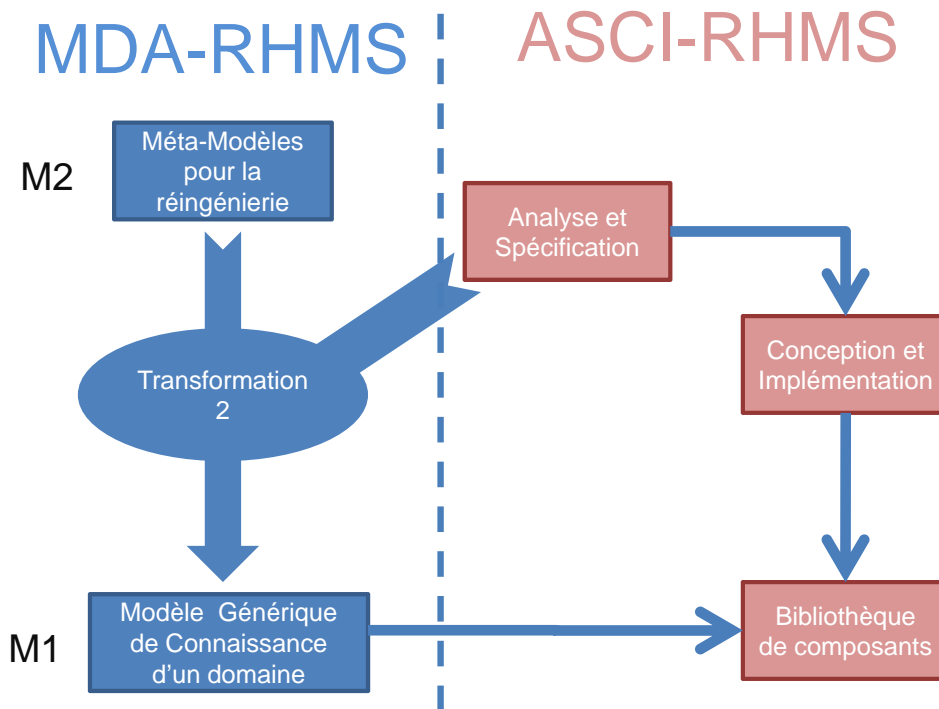


FIGURE 5.11 – Passage du Méta-Modèle au Modèle Générique de Connaissance d'un domaine

Cette étape est la plus importante de notre méthodologie et permettra d'isoler les composants, c'est-à-dire que nous devons isoler les différents types d'holons qui doivent intervenir dans la réingénierie et ainsi détailler leurs relations.

Le **MGC** du domaine ainsi réalisé peut être instancié sur tout système de ce domaine. Le **MGC** est donc composé des deux modèles de niveau M1 conformément aux deux méta-modèles présentés plus haut dans ce chapitre, plus la décomposition systémique venant de **ASCI**.

A partir du **MGC** et des étapes de Conception et d'Implémentation, une bibliothèque de composants du domaine est construite. L'élaboration d'une bibliothèque de composants va consister dans un premier temps à filtrer et faire dériver les classes du **MGC** du domaine. Ces classes dérivent après avoir été filtrées pour ne garder que celles qui seront utiles à la construction de modèles d'action. D'autres « classes techniques » peuvent également être construites pour le besoin de la programmation [Sarramia 02]. L'étape de Conception consiste à formaliser les composants génériques sans prendre en compte les spécificités de telle ou telle implémentation (langage de simulation, langage de programmation. . .). L'étape d'Implémentation consiste à coder ces composants dans un langage souhaité afin qu'ils puissent être utilisés et réutilisés dans les différents modèles d'action qui seront développés pour chaque système étudié.

5.7.3 Utilisation à partir du MGC d'un domaine

L'utilisation de la méthodologie de réingénierie à partir du **MGC** d'un domaine correspond au processus de modélisation pour un système étudié d'un domaine dont on disposerait déjà de son **MGC**. La figure 5.12 explique les différentes étapes qui sont préconisées par **ASCI-RHMS**.

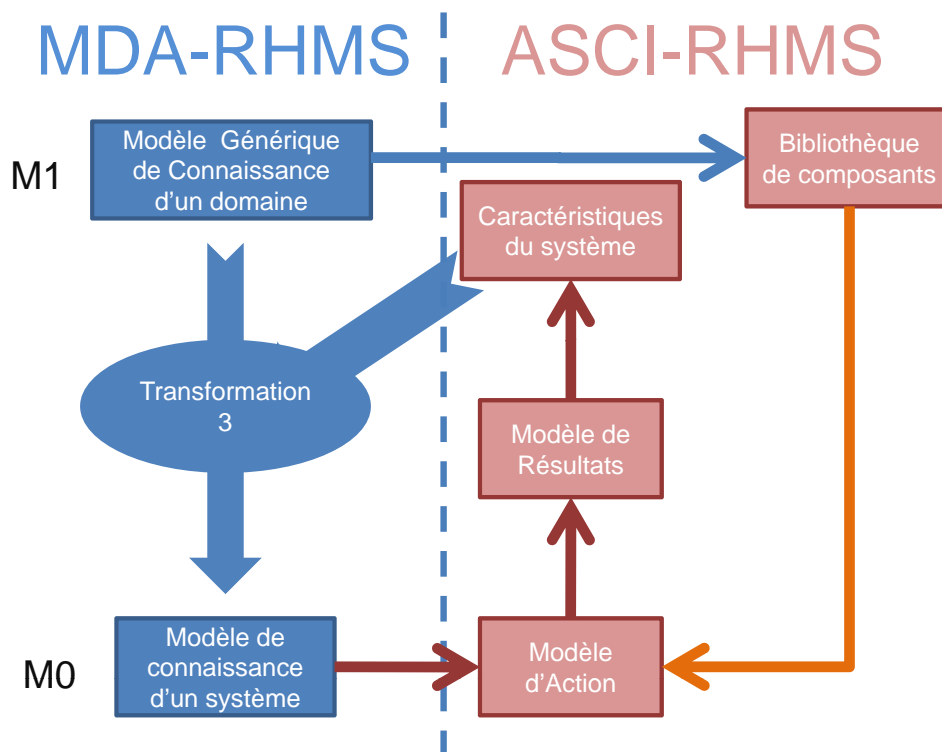


FIGURE 5.12 – Passage du Modèle Générique de Connaissance d'un domaine au Modèle de Connaissance d'un système du domaine

Ce processus est généralement itératif et composé de quatre étapes que l'on peut voir sur la figure 5.12 et sur la figure 5.10 qui sont [Rodier 10] :

- l'élaboration d'un *Modèle de Connaissance (MC)* d'un système,
- l'obtention d'un modèle d'action à partir du modèle de connaissance,
- l'exploitation du modèle d'action pour évaluer les performances du système,
- l'interprétation des résultats avec la déduction des actions à effectuer sur le système.

La construction du modèle de connaissance doit être réalisée en étroite collaboration avec les experts du domaine. Sa construction consiste en le recueil et la formalisation de la connaissance sur le système étudié. Il est donc très important de disposer de méthodes et d'outils compréhensibles par les experts du domaine tel que le formalisme JIS Z 8206 (cf section 3.5.4) pour exprimer les caractéristiques du système. Un exemple d'utilisation d'un tel outil est donné au chapitre 7. Nous aboutirons alors au *MC* d'un système du domaine. Ce *MC* sera composé d'éléments instanciés du *MGC* du domaine dont une instanciation des diagrammes des *ED* (passage du niveau M1 à Mo). Par conséquent, ces diagrammes représenteront un système tel qu'il est actuellement, c'est-à-dire non contrôlé par le produit et le système tel que nous souhaiterions qu'il soit, c'est-à-dire contrôlé par le produit. Ces diagrammes permettront plus facilement la réingénierie d'un système de production.

L'instanciation des *ED* du *MGC* du domaine nous montrera de façon détaillée les composants à étudier et à reprogrammer pour permettre le déploiement du contrôle par le produit au niveau du système. Nous pourrions prendre dans la bibliothèque de composants, ceux qui correspondront à une construction du système contrôlé par le produit.

5.8 CONCLUSION

Nous avons décrit dans ce chapitre une méthodologie de réingénierie pour le contrôle par le produit de système manufacturier, basée sur les concepts les plus récents : le paradigme holonique, le langage **SysML**, une démarche **MDA**, l'approche par composant et une version adaptée de la méthodologie **ASCI** que nous avons appelée **ASCI-RHMS**. **ASCI** a été adaptée pour lui permettre d'être utilisée avec l'approche par composant et de permettre une gestion de composant avec une partie logicielle et une partie physique. Nous nous sommes inspirés ainsi de la méthodologie **MBCSA** qui utilise l'approche par composant et de la démarche **MDA** pour compléter **ASCI**. La réingénierie s'effectue en mettant en place des règles de transformation pour passer d'un méta-modèle à un autre. Les modèles sont écrits à partir du langage **SysML**. Le Méta-Méta-Modèle est décrit avec le *Block Definition Diagram - diagramme de définition de bloc (BDD)*, permettant une définition et une description aisées des éléments du système (bloc en **SysML**); les Méta-Modèles sont décrits avec des *Internal Block Diagram - diagramme interne de bloc (IBD)*, pouvant aisément décrire une structure suivant les interactions souhaitées entre les blocs. Nous préconisons le même type de diagramme pour les derniers niveaux de modélisation. Cette méthodologie **ASCI-RHMS** doit être en relation avec les transformations effectuées dans la démarche **MDA**. La démarche complète avec la démarche **MDA** à 4 niveaux (**MDA-RHMS**) et la méthodologie **ASCI-RHMS** pour faire la liaison entre les derniers niveaux constitue la *Méthodologie de Réingénierie des HMS (MRHMS)*. Les interactions entre ces deux méthodologies (**MDA-RHMS** et **ASCI-RHMS**) sont expliquées sur la figure 5.13.

Sur cette figure, les quatre niveaux d'abstraction décrits dans ce chapitre sont représentés. Le lien entre les derniers niveaux est assuré par la méthodologie **ASCI-RHMS** qui permet de passer du méta-modèle à un domaine (classe de système), puis du domaine à un système du domaine. Nous proposons d'utiliser cette méthodologie soit à partir du niveau M2 pour la construction d'un *Modèle Générique de Connaissance (MGC)* d'un domaine, dans le cas où l'on souhaite étudier la réingénierie d'un domaine, soit à partir du niveau M1 pour l'élaboration d'un *Modèle de Connaissance (MC)* d'un système pour étudier la réingénierie d'un système d'un domaine dont l'on dispose déjà du **MGC**, soit en utilisant cette méthodologie avec ces deux possibilités pour effectuer une étude complète d'un système avec son domaine. Dans tous les cas, l'étude de la réingénierie s'effectue en isolant les composants utilisés lors d'un fonctionnement contrôlé par les ressources. Notre méthodologie explicite les relations qui doivent être mises en œuvre entre eux pour permettre un contrôle (intropérabilité et pilotage) par le produit. Le Méta-Méta-Modèle du niveau M3 pourrait aussi être réutilisé pour la construction d'autres Méta-Modèles pour une autre réingénierie que pour le contrôle par le produit ou encore pour un autre type de modélisation correspondant à cette vision d'un système de production (les Méta-Modèles devront toujours être conformes au Méta-Méta-Modèle).

Nous proposons une application dans le prochain chapitre sur un domaine : le circuit du médicament dans un hôpital. Nous montrerons la mise en pratique de la *Méthodologie de Réingénierie des HMS (MRHMS)* décrite dans ce chapitre.

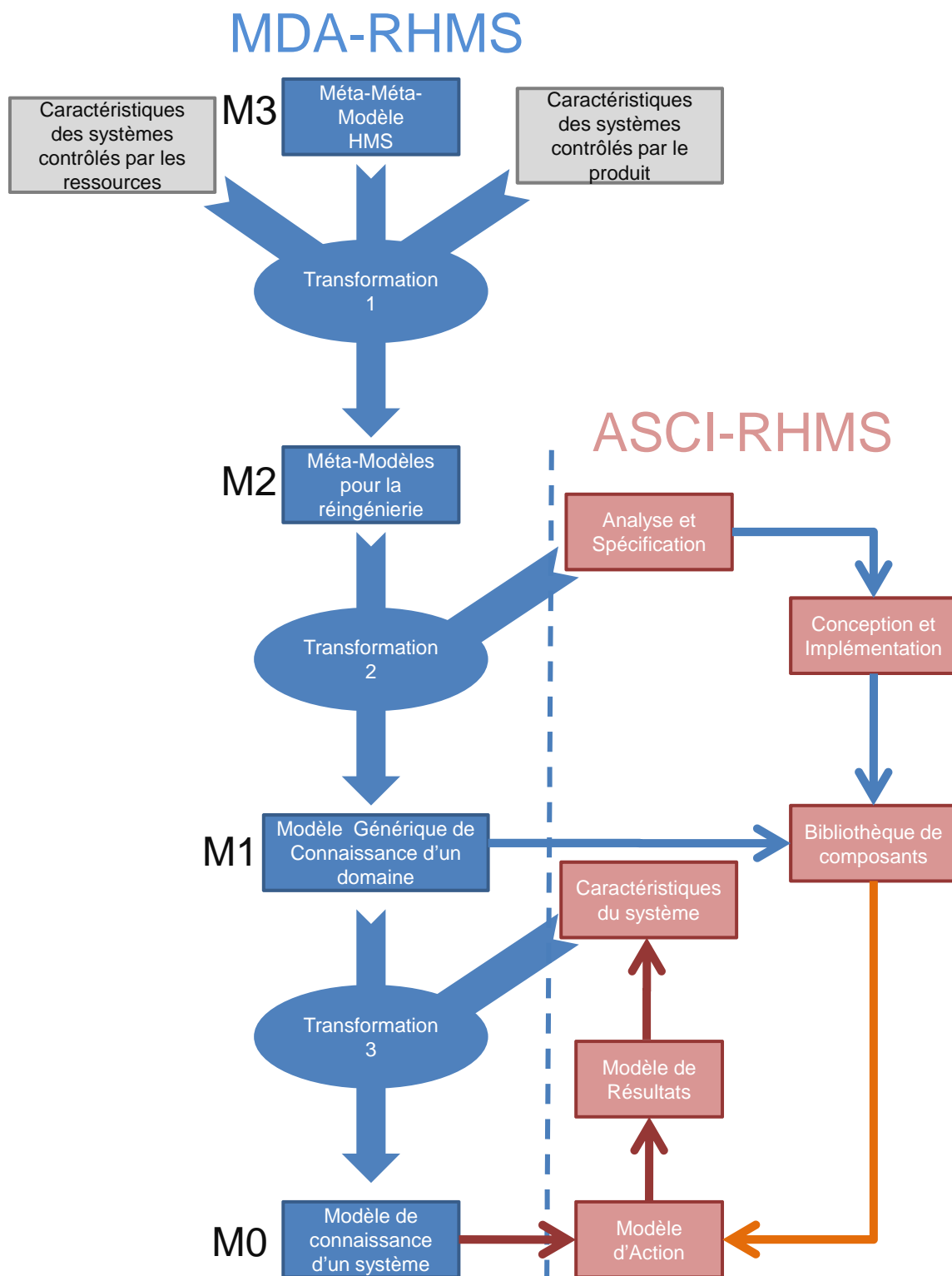


FIGURE 5.13 – MRHMS et ses interactions méthodologiques internes entre MDA-RHMS et ASCI-RHMS

6

INSTANCIATION ET APPLICATION AU DOMAINE DU CIRCUIT DU MÉDICAMENT

RÉSUMÉ DU CHAPITRE 6

Nous faisons l'analogie entre le circuit du médicament et un système de production, et plus précisément avec un système d'assemblage. Nous avons proposé une méthodologie de réingénierie d'un système de production : MRHMS. Notre méthodologie a plusieurs finalités :

- Elle peut être utilisée du « Méta-modèle pour la réingénierie » au « Modèle de connaissance d'un système » ; il s'agit alors d'une étude complète (nous entendons par étude complète l'étude du domaine + l'étude d'un système bien particulier de ce domaine) ;*
- L'analyste peut n'avoir qu'à étudier uniquement un domaine sans s'intéresser à un système particulier ; il pourra alors s'appuyer sur notre méthodologie (cette partie est l'objet de ce chapitre) ;*
- Si une étude préalable s'est déjà intéressée au domaine du système, notre méthodologie peut alors poursuivre cette étude vers l'étude du système particulier (ce qui sera l'objet du chapitre 7).*

Ce chapitre est un cas d'étude du circuit du médicament par cette méthodologie, nous créons ici le Modèle Générique de Connaissance d'un circuit du médicament. La structure de ce chapitre suit les étapes préconisées par ASCI-RHMS, c'est-à-dire :

- Analyse avec une décomposition systémique et les diagrammes des Entités Décisionnelles pour expliquer la structure et les relations entre les composants du domaine,*
- Spécification avec des diagrammes d'activité pour représenter son comportement,*
- Conception et Implémentation avec plusieurs exemples : Analyse du circuit chariot, un logiciel de validation au lit du malade et un logiciel de simulation (Automod).*

Nous détaillons ainsi la création d'un Modèle Générique de Connaissance à partir des deux premières étapes et la création d'une bibliothèque de composants réutilisables à partir des deux dernières.

SOMMAIRE

6.1	INTRODUCTION	98
6.2	PHASE 1 - ANALYSE	100
6.2.1	ASCI pour le pilotage hybride	100
6.2.2	Sous-Système Physique (SSP)	101
6.2.3	Sous-Système Logique (SSL)	103
6.2.4	Sous-Système Décisionnel (SSD)	103
6.3	PHASE 1 - SPÉCIFICATION	105
6.3.1	Diagramme d'activité générale	105
6.3.2	Diagramme d'activité représentant la préparation d'une livraison	106
6.3.3	Diagramme d'activité représentant la livraison et le rangement d'une livraison de doses de médicament	107
6.4	PHASE 1 - ENTITÉS DÉCISIONNELLES ET TRANSFORMATION 2 DE LA DÉMARCHE MDA-RHMS	109
6.4.1	Le SIH	109
6.4.2	Les diagrammes des entités décisionnelles	109
6.4.2.1	Isolation des holons	109
6.4.2.2	Représentation d'une dispensation globale et globalisée	110
6.4.2.3	Représentation d'une dispensation globale et globalisée	111
6.5	PHASE 1 - BILAN	112
6.5.1	La démarche de réingénierie d'un circuit du médicament	112
6.5.2	Création d'une bibliothèque de composants	113
6.6	PHASE 2 - CONCEPTION ET IMPLÉMENTATION : CAS D'ÉTUDE D'UNE INTERFACE DE LOGICIEL DE SIMULATION	113
6.6.1	Conception	113
6.6.2	Implémentation	115
6.7	PHASE 2 - CONCEPTION ET IMPLÉMENTATION : CAS D'ÉTUDE DU SUIVI DU CIRCUIT DU MÉDICAMENT PAR IDENTIFICATION RADIOFRÉQUENCE (RFID)	116
6.7.1	Cahier des charges	116
6.7.2	Conception	118
6.7.3	Implémentation	121
6.7.3.1	Choix des logiciels de programmation	121
6.7.3.2	Préparation	122
6.7.3.3	Administration	123
6.8	PHASE 2 - CONCEPTION ET IMPLÉMENTATION : CAS D'ÉTUDE DE L'UTILISATION DU LOGICIEL DE SIMULATION AUTO MOD	124
6.8.1	État de l'art sur la simulation du circuit du médicament	124
6.8.2	Bilan et perspectives sur la simulation du CDM	125
6.8.3	Le logiciel de simulation Automod	125
6.8.3.1	Présentation de Automod	125
6.8.3.2	Resources	126
6.8.3.3	Load	126
6.8.3.4	Queue	126
6.8.3.5	Système de déplacement	126
6.8.3.6	Order list	127
6.8.4	Simulation d'une pharmacie générique	127

6.8.4.1	Pharmacie générique pour une livraison globale	127
6.8.4.2	Pharmacie générique pour une livraison nominative	128
6.8.4.3	Représentation des stocks de médicaments	130
6.8.4.4	Interface d'entrée-sortie	130
6.9	PHASE 2 - BILAN	131
6.10	GÉNÉRICITÉ DE L'APPROCHE PROPOSÉE	131
6.11	CONCLUSION	132

6.1 INTRODUCTION

Le *Circuit Du Médicament* (CDM) dans un hôpital commence par l'écriture de la prescription; il consiste ensuite en la préparation et l'envoi de doses de médicament pour chaque patient chaque jour, et enfin à l'administration des médicaments (cf chapitre 1). La réingénierie du CDM est un enjeu important pour réduire les erreurs médicamenteuses et les coûts dans le milieu hospitalier. Ce chapitre montre une application de la méthodologie MRHMS présentée au chapitre précédent sur le domaine du CDM. Le positionnement du chapitre 6 par rapport à la méthodologie MRHMS est montré sur la figure 6.1.

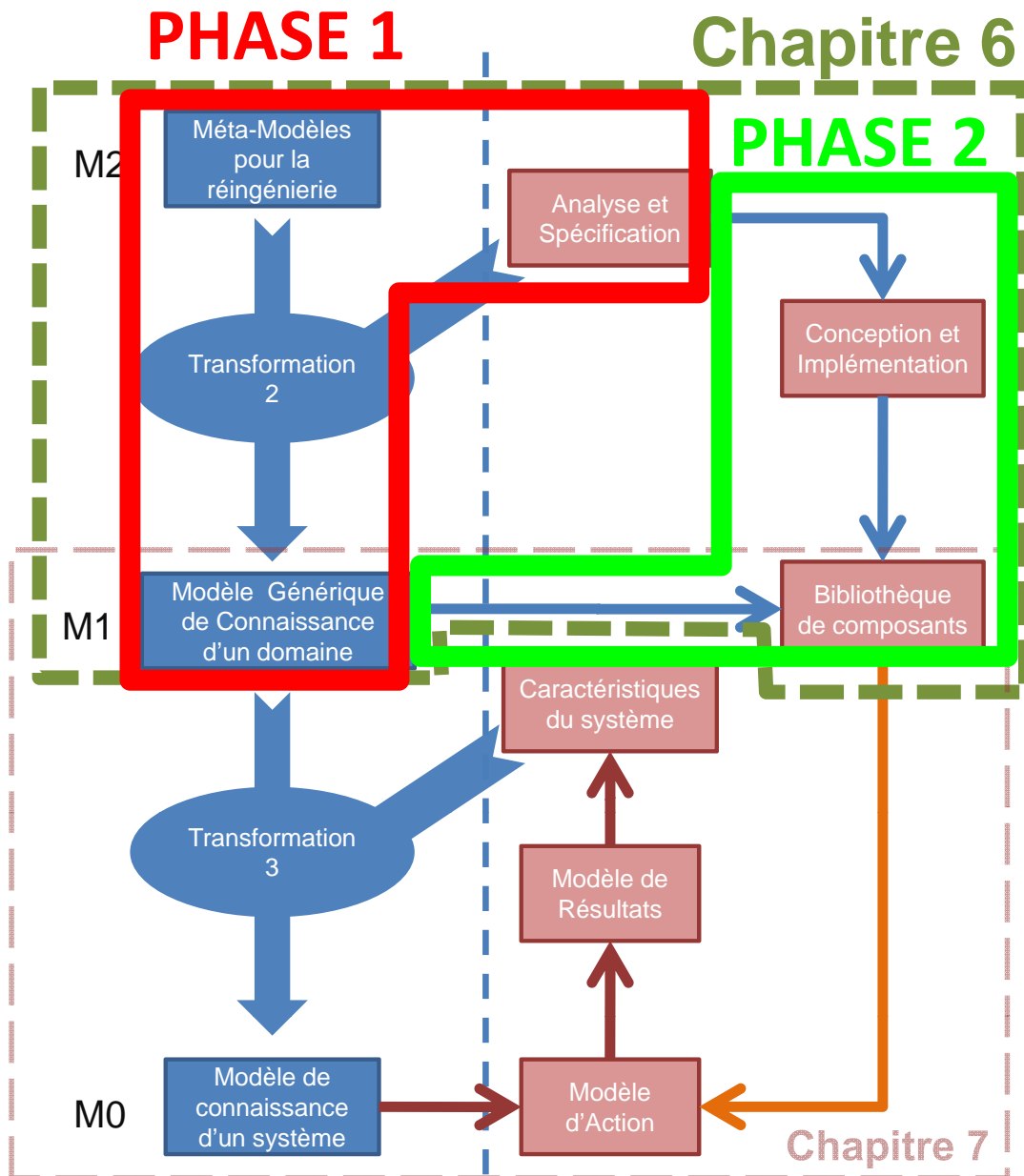


FIGURE 6.1 – Positionnement du chapitre 6 dans le mémoire

Cette méthodologie nous permet de modéliser les interactions avec le **SIH** pour aboutir ainsi à une réorganisation plus aisée. Nous expliquerons en quoi nous pouvons voir le **CDM** comme un **HMS**.

Cette modélisation du **CDM** permet de prendre en compte toutes les possibilités de dispensation. Elle montre que les trois actes essentiels à toute dispensation sont la préparation, la livraison et le rangement. Cette modélisation permet d'obtenir la base de connaissance à partir de laquelle on peut simuler le **CDM**, c'est-à-dire évaluer les performances d'une configuration par rapport à une autre. La vérification et la validation du modèle sont effectuées lors de réunions avec les pharmaciens du **CHU** de Clermont-Ferrand. Nous nous sommes aussi inspiré des travaux d'autres hôpitaux français comme Annecy [Majchrzak 06], Lyon ou Grenoble [Bedouch 09]. L'objectif est de pouvoir prendre en compte et modéliser de façon précise tous les modes de fonctionnement en se basant sur ces différentes expériences.

Ce chapitre commencera par présenter les étapes de la phase 1 de la figure 6.1 préconisées par **ASCI** pour la création d'un *Modèle Générique de Connaissance* (**MGC**) en l'appliquant au **CDM**, c'est-à-dire l'étape d'Analyse (section 2), l'étape de Spécification (section 3) et enfin les diagrammes des *Entités Décisionnelles* (**ED**) (section 4). Nous ferons un bilan de cette phase et nous expliquerons comment nous construisons une bibliothèque de composants réutilisables (section 5), puis nous illustrerons les étapes de Conception et d'Implémentation de la phase 2 qui permettent d'enrichir cette bibliothèque :

- Section 6 avec la proposition d'une interface d'entrée/sortie pour des logiciels de simulation,
- Section 7 avec la proposition d'un dispositif de suivi du **CDM** par Identification Radiofréquence (RFID),
- Section 8 avec l'utilisation du logiciel de simulation Automod.

Nous avons résumé, sur la figure 6.2, le plan de ce chapitre en détaillant les sections sur les cas d'étude de composants présentés dans la deuxième phase .

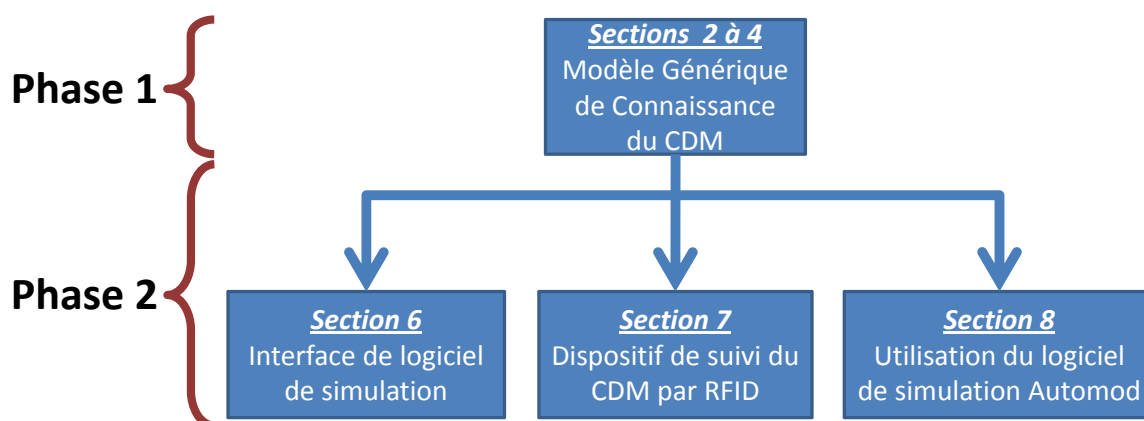


FIGURE 6.2 – Construction d'une bibliothèque de composants réutilisables

6.2 PHASE 1 - ANALYSE

6.2.1 ASCI pour le pilotage hybride

Nous avons effectué une décomposition systémique à partir des sous-systèmes que préconise la méthodologie **ASCI**. Nous utilisons le langage **UML** [OMG 10b] pour modéliser le circuit du médicament. Il a été montré dans le chapitre 1 que le **CDM** peut être vue comme un système d'assemblage. [El Haouzi 08a, El Haouzi 08b] présentent une décomposition systémique d'un système d'assemblage en utilisant comme cadre méthodologique *ASCI pour le Pilotage Hybride* (**ASCI-PH**); ils cherchent à obtenir un pilotage combinant un pilotage décentralisé et centralisé pour permettre un maximum de flexibilité. En reprenant ces travaux et en l'appliquant à notre méthodologie **ASCI-RHMS**, nous avons effectué une décomposition systémique du **CDM**. Nous appelons notre méthodologie **ASCI-RHMS** instanciée sur le domaine du **CDM**, la méthodologie *ASCI pour la Réingénierie du CDM* (**ASCI-CDM**). Nous expliquons notre démarche de modélisation en montrant la correspondance entre la méthodologie **ASCI-PH** et la méthodologie **ASCI-CDM** sur la figure 6.3.

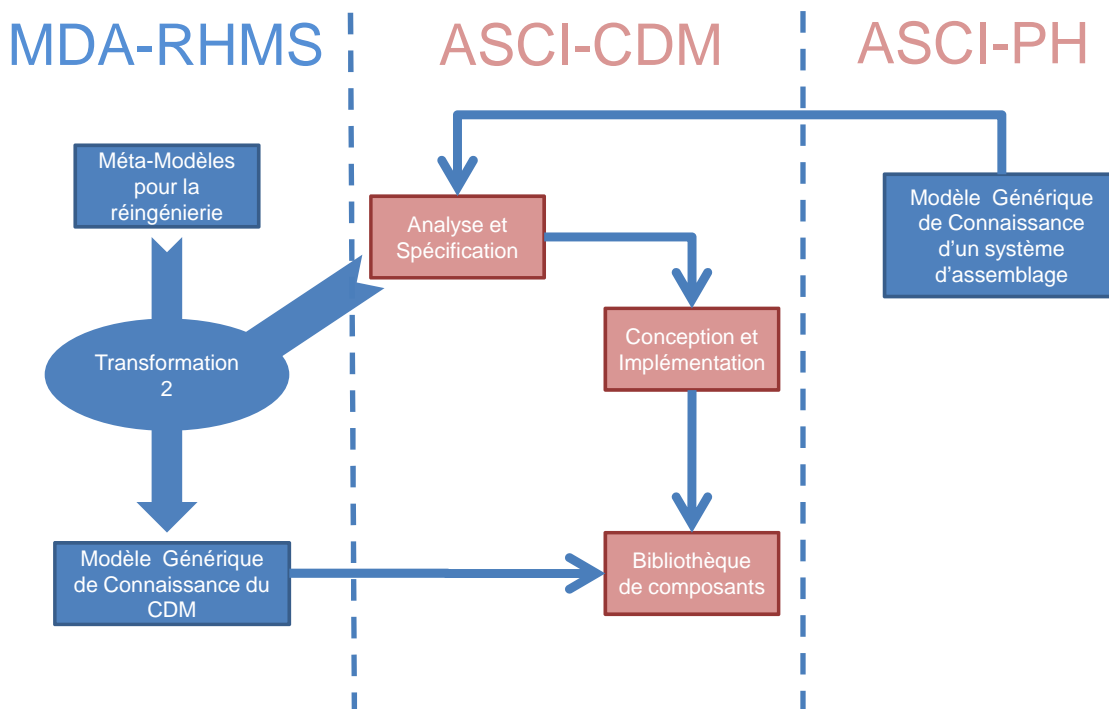


FIGURE 6.3 – Démarche de modélisation pour la conception d'un MGC du CDM

6.2.2 Sous-Système Physique (SSP)

le *Sous-Système Physique* (SSP) définit l'ensemble des moyens physiques (pour la production, le stockage, la manutention et le transport), leur répartition géographique et leur interconnexion. L'élément principal du SSP est la ligne d'assemblage, composée d'un ou plusieurs postes de charge. Un poste de charge est composé d'un ou plusieurs postes de travail, alimentés par un ou plusieurs « feeders ». En effet, pour alimenter ce flux, [El Haouzi 08a] explique que divers sous-ensembles sont nécessaires ; ces sous-ensembles sont fabriqués sur des lignes appelées « feeders » ou des cellules indépendantes. Ces « feeders » viennent alimenter la ligne principale. L'assemblage d'un produit peut nécessiter des composants assemblés sur ces « feeders », de sorte que l'assemblage final du produit nécessite l'attente de tous les composants. Un problème de synchronisation des différentes lignes en amont de la ligne dite principale peut apparaître. Pour le CDM, ces « feeders » peuvent être des lignes appliquées à différents types de médicaments, comme par exemple des *Dispositifs Médicaux Stériles* (DMS) ou des solutés (médicaments sous forme liquide).

Un poste de travail peut être composé par des robots. Il existe des robots pour faire de la préparation globale et d'autres pour de la préparation nominative. Un poste de travail peut se situer à la pharmacie ou dans la salle de préparation dans les unités de soins. Ils sont reliés entre eux par un moyen de manutention. Nous avons détaillé ce sous-système en reprenant et en augmentant le diagramme de classe écrit par [El Haouzi 08a, El Haouzi 08b]. Les classes extraites de leurs diagrammes sont représentées en grisé. Le SSP est représenté par le diagramme de classes de la figure 6.4.

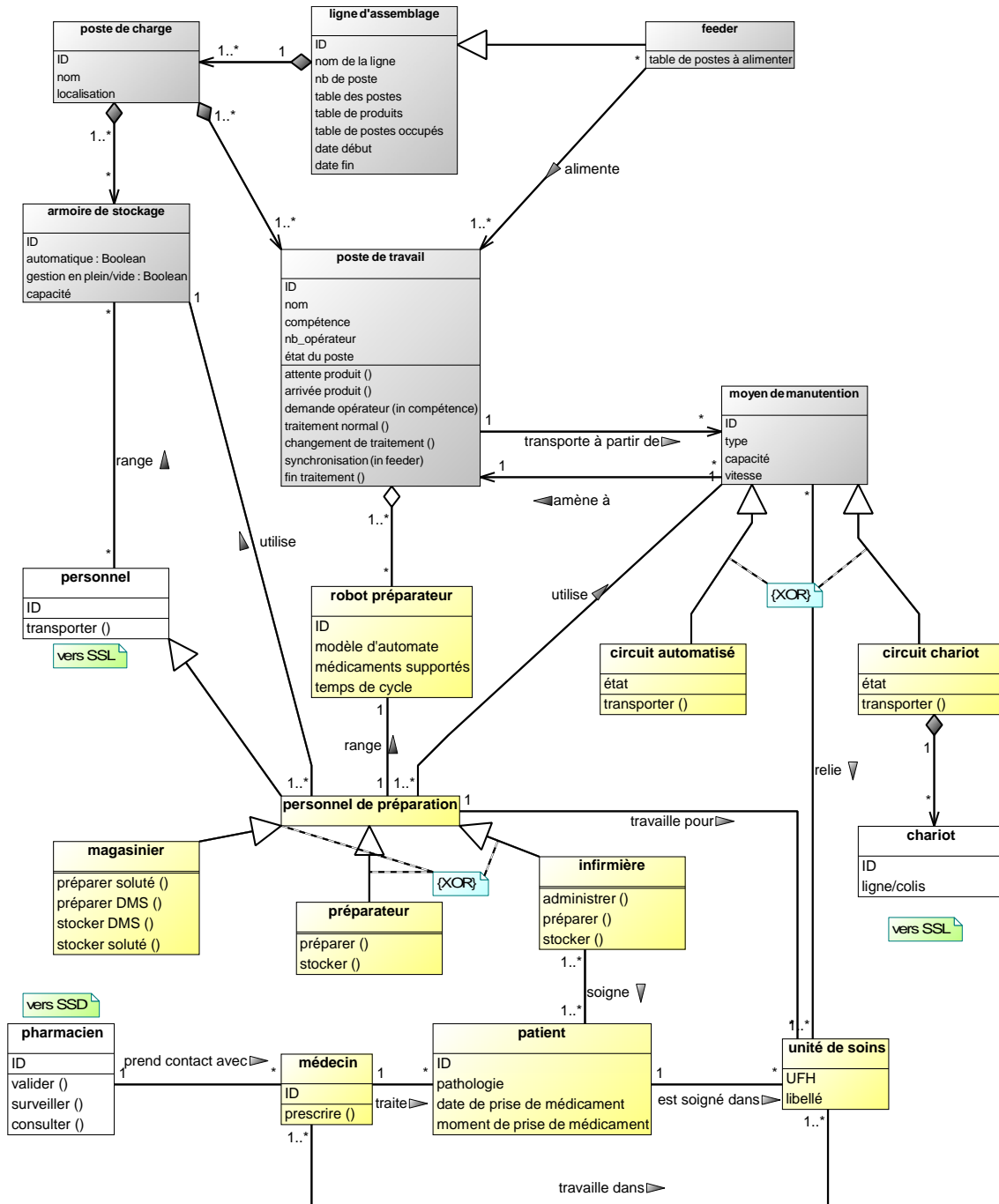


FIGURE 6.4 – Sous-Système Physique de ASCI-CDM

6.2.3 Sous-Système Logique (SSL)

le *Sous-Système Logique (SSL)* est constitué des entités que le système doit traiter, des ensembles d'opérations concernant ces flux, ainsi que des nomenclatures qui s'y rapportent. La pharmacie livre dans l'unité de soins les *Doses Préparées pour un Patient (DPP)* si le fonctionnement est en **DJIN**. Si la dispensation est globale ou globalisée, la pharmacie livre la préparation des dotations, appelées les *Doses Préparées pour un Service (DPS)* (livraison de « pièces » pour un assemblage dans le service). Le **SSL** décrit les éléments spécifiant et caractérisant le flux de produit : sa gamme, sa nomenclature (prescription), le programme directeur de production qui contient les informations sur le nombre de produits à fabriquer pour une période déterminée, l'ordre du passage, la date du début de fabrication de chaque produit. Le **SSL** est centré sur le produit, c'est-à-dire sur les *Doses Préparées pour un Patient (DPP)* pour le **CDM**. [El Haouzi 08a] explique que le Programme Directeur de Production (PDP) a pour objectifs d'une part, le séquençement des ordres de fabrication en fonction des règles préétablies et l'analyse des messages d'exception et, d'autre part, de valider les séquençements et de s'assurer que le lancement des dossiers de fabrication se déroulera normalement. Les déclencheurs de la fonction de planification sont essentiellement les commandes clients complètes. Pour nous, les « clients » du monde industriel correspondent au médecin prescrivant une ordonnance à un patient dans un établissement hospitalier. Le **SSL** est représenté par le diagramme de classe de la figure 6.5 avec en grisé, les éléments repris de [El Haouzi 08a].

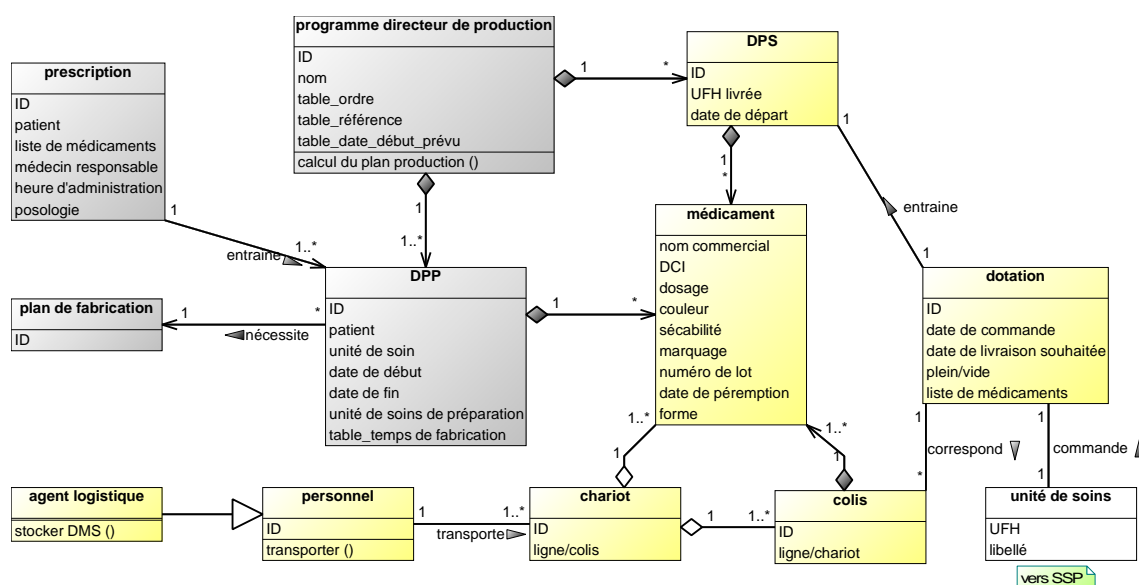


FIGURE 6.5 – Sous-Système Logique de ASCI-CDM

6.2.4 Sous-Système Décisionnel (SSD)

le *Sous-Système Décisionnel (SSD)* contient les règles de gestion et de fonctionnement du système. Les centres de décision centralisés concernent les trois niveaux : stratégique, tactique, et opérationnel. Les décisions centralisées sont prises par les pharmaciens qui sont responsables du bon déroulement du **CDM**. Les décisions décentralisées sont prises par les éléments contenant de l'intelligence au plus près du terrain : les **DPP** équipées de puces **RFID**, les robots-préparateurs, le circuit chariot et le circuit automatisé avec leurs **SI** correspondant et

le personnel de préparation qui gère ces SI. Le SSD est composé de deux types de centres de décision : centralisés et distribués. Il est représenté sur la figure 6.6 avec, en grisé, les éléments repris de [El Haouzi 08a].

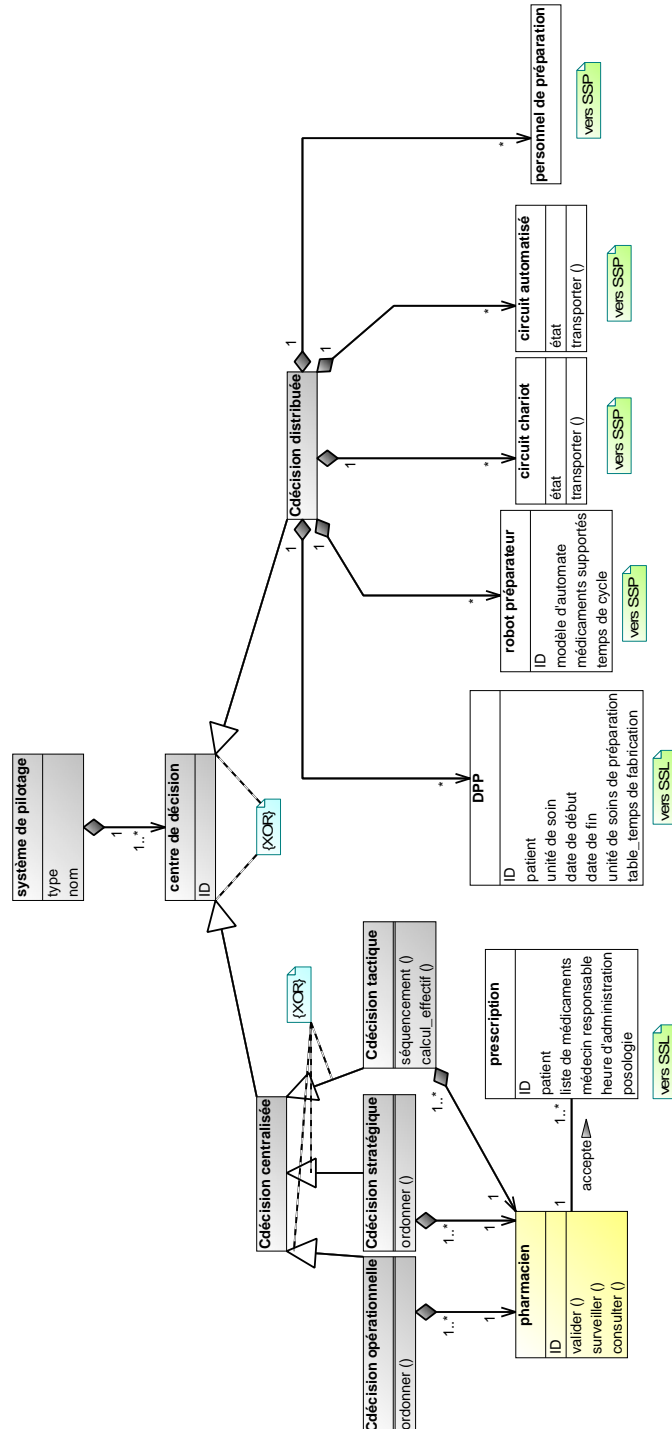


FIGURE 6.6 – Sous-Système Décisionnel de ASCI-CDM

6.3 PHASE 1 - SPÉCIFICATION

6.3.1 Diagramme d'activité générale

Le fonctionnement global d'un CDM peut être résumé avec la figure 6.7.

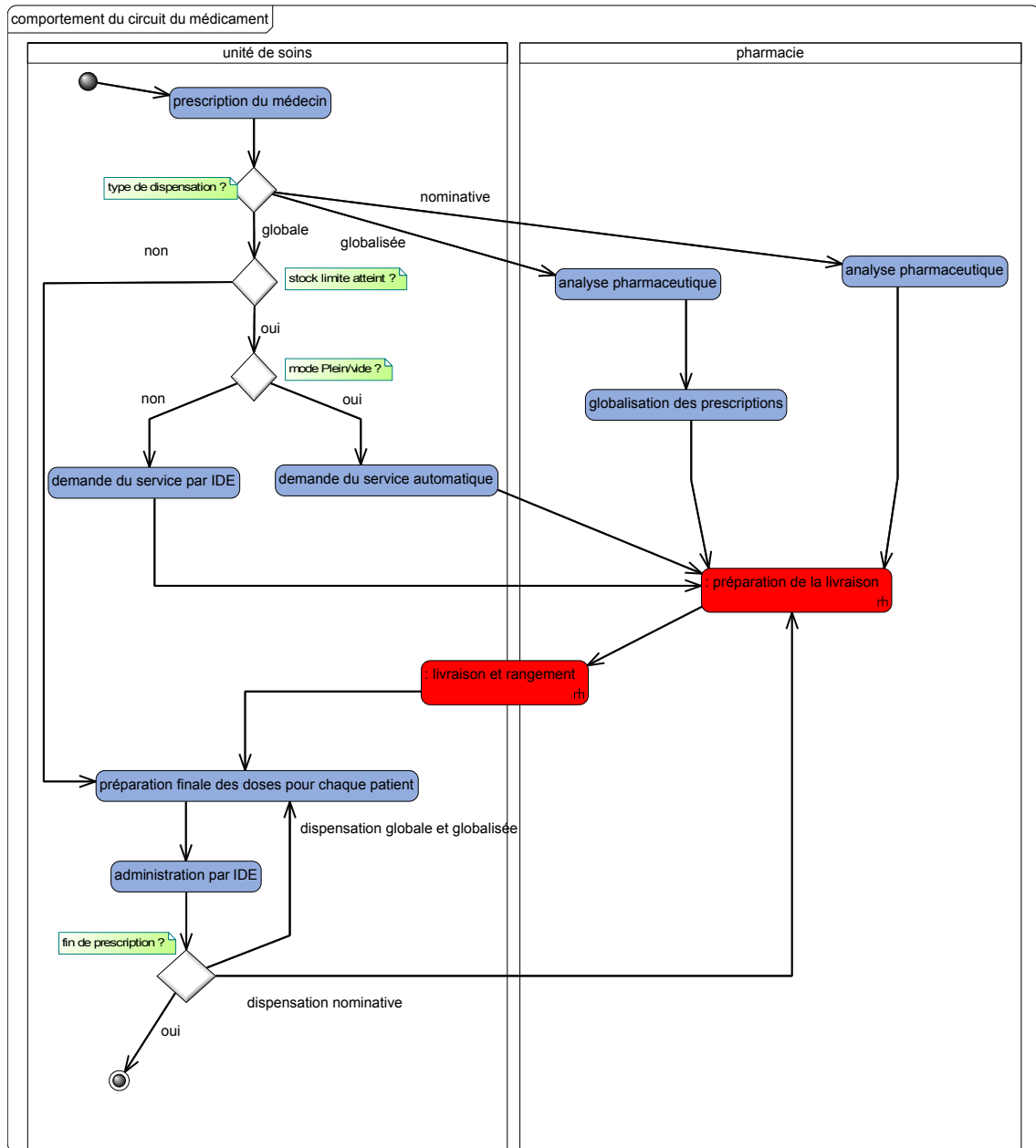


FIGURE 6.7 – Diagramme d'activité représentant le comportement général du circuit du médicament

Il décrit le comportement quel que soit le mode de fonctionnement : de l'écriture de la prescription par le médecin dans l'unité de soins à l'administration des médicaments par l'Infirmier Diplômé d'État (IDE) au patient. Dans ce diagramme, deux activités sont composites, c'est-à-dire qu'elles sont elles-mêmes décomposées en un autre diagramme d'activité. Cela concerne l'activité « préparation de la livraison » et l'activité « livraison et rangement ».

6.3.2 Diagramme d'activité représentant la préparation d'une livraison

Toutes les possibilités pour la préparation d'une livraison de médicament sont représentées sur le diagramme de la figure 6.8.

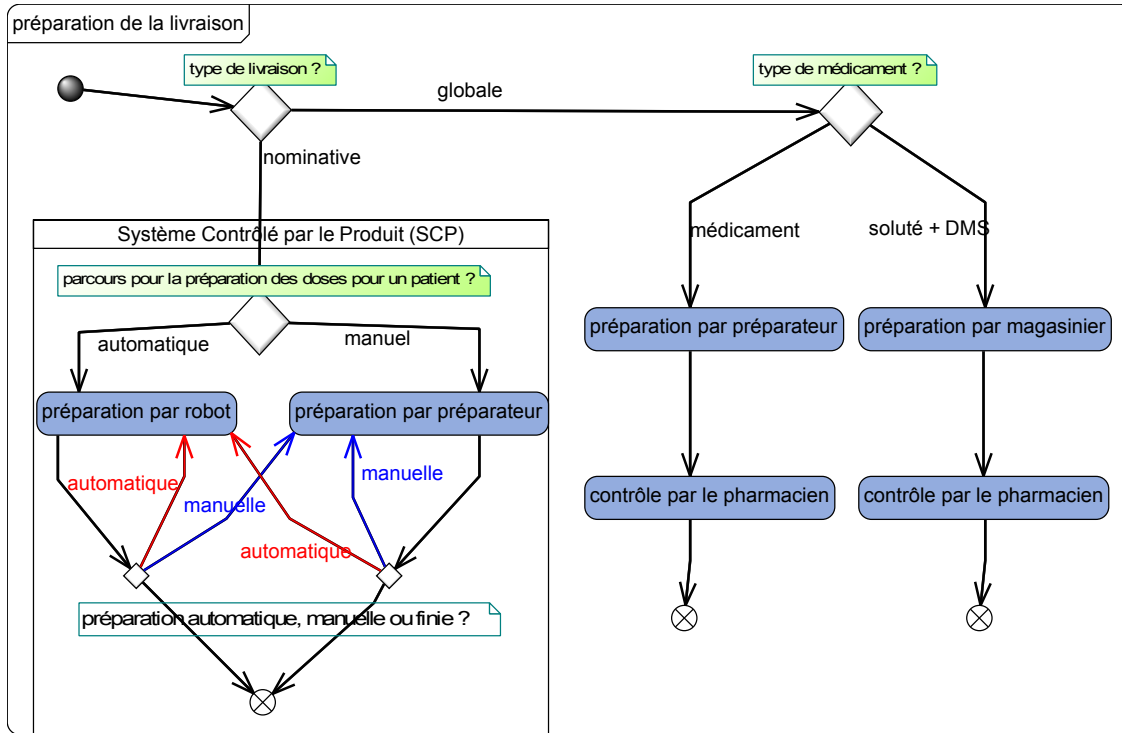


FIGURE 6.8 – Diagramme d'activité représentant une préparation de doses

On appelle « livraison globale », la dispensation globale et globalisée. La préparation d'une livraison correspond à la préparation effectuée à la pharmacie, c'est donc soit l'ensemble des *Doses Préparées pour un Patient* (DPP) en livraison nominative, soit l'ensemble des *Doses Préparées pour un Service* (DPS) pour une livraison globale. Ce diagramme correspond à l'activité composite « Préparation d'une livraison ». Quand la livraison est nominative, on retrouve le concept de SCP qui permet de piloter ce type de livraison. Par exemple, dans le cas de plusieurs préparations pour un poste de travail équipé de robots, le parcours rouge est emprunté (préparation automatique) et en cas de plusieurs préparations sur un poste de travail avec un préparateur, le parcours bleu est emprunté (préparation manuelle). Une fois la préparation finie, on peut sortir de l'activité composite et passer à l'activité suivante qui est l'activité composite « livraison et rangement ».

6.3.3 Diagramme d'activité représentant la livraison et le rangement d'une livraison de doses de médicament

Nous avons récapitulé ce qui est autorisé dans les établissements de santé pour chaque acteur du CDM et pour chaque type de médicament dans le tableau 6.1.

	Livraison	Rangement médicament	Rangement soluté	Rangement DMS
Préparateur	X	X	X	X
Magasinier	X		X	X
Agent log	X			X
IDE		X	X	X

TABLE 6.1 – Actions autorisées pour chaque acteur

Par exemple, le magasinier sera autorisé à livrer tout type de médicament mais sera autorisé à ranger seulement les solutés et les DMS. L'IDE ne livre pas mais peut ranger tous les types de médicament. Cette analyse préliminaire nous permet de créer un arbre décisionnel représentant toutes les possibilités de fonctionnement. La figure 6.9 représente cet arbre décisionnel et explique tous les modes de fonctionnement possibles pour la livraison et le rangement des produits pharmaceutiques dans les armoires de service. Elle correspond à l'activité composite « livraison et rangement » de la figure 6.7.

Chaque sortie de cet arbre est une solution complète de rangement et de livraison possible. Chaque livraison correspondra à une branche de cet arbre.

6.4 PHASE 1 - ENTITÉS DÉCISIONNELLES ET TRANSFORMATION 2 DE LA DÉMARCHE MDA-RHMS

6.4.1 Le SIH

Nous avons expliqué au chapitre 1 que le **SIH** a beaucoup de similitudes avec un **SIE**. En effet, le logiciel utilisé à la pharmacie peut être comparé à un **ERP**. L'organisation d'un **SIE** et les concepts décrits pour un *Système de Systèmes* (**SdS**) sont valables pour un **SIH**. Plus précisément, plusieurs logiciels indépendants avec différentes missions doivent collaborer entre eux : nous rencontrons la même problématique d'interopérabilité que dans le monde industriel.

Le **SIH** est composé de plusieurs **SI** correspondant à chaque service d'un hôpital comme dans une entreprise où il peut y avoir des **SI** propres à chaque service. Nous avons représenté le **SIH** sur la figure 6.10. L'acronyme *Unité Fonctionnelle* (**UF**) représente le service d'un hôpital. Nous avons montré quelques exemples de **SI** comme par exemple le *Système d'Information Pharmaceutique* (**SIP**) ou le **SI** d'un service **UF_x** ou **UF_y**.

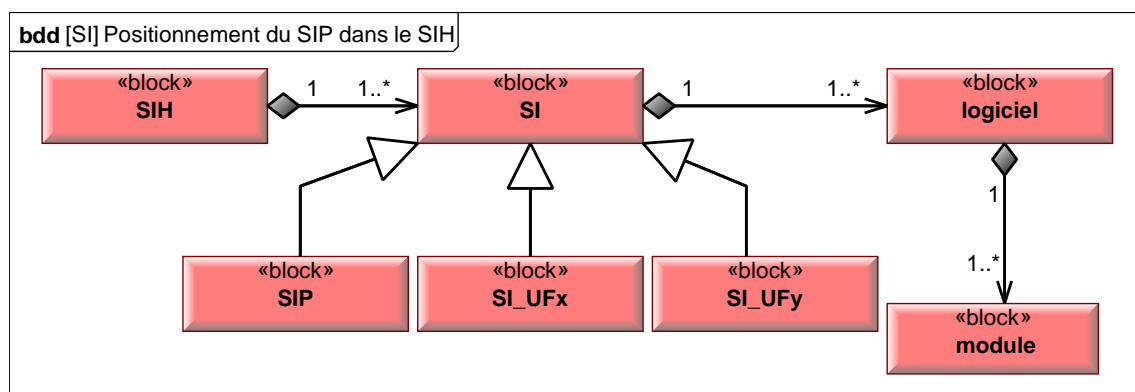


FIGURE 6.10 – Place du Système d'Information Pharmaceutique dans le Système d'Information Hospitalier

C'est dans ce contexte que nous comparons le produit et les *Doses Préparées pour un Patient* (**DPP**). Ces **DPP** sont composées de parties physiques et informationnelles dans le cas d'utilisation de puces **RFID**. Cette partie informationnelle communique avec le **SIH** et les ressources utilisées.

6.4.2 Les diagrammes des entités décisionnelles

6.4.2.1 Isolation des holons

[El Haouzi 08b] propose d'enrichir la décomposition systémique classique avec un diagramme représentant les entités décisionnelles qui font le lien entre les sous-systèmes physique et décisionnel. Nous avons repris cette idée dans la méthodologie **ASCI-RHMS**. Nous proposons d'adapter cette représentation avec une instanciation des deux diagrammes présentés au niveau M2.

A partir de l'analyse effectuée à la section précédente, nous pouvons isoler les holons de notre système :

- les **DPP** correspondent au **HP**,
- les circuits automatisés et chariots, les robots préparateurs ainsi que le personnel de

préparation gérant les armoires de stockage (cueillette manuelle) correspondent aux **HR**,

- les pharmaciens qui assurent les décisions centralisées (qui supervisent le bon fonctionnement de tout le circuit) seront considérés comme les **HS**.

Les relations entre les holons et les autres éléments du **CDM** sont présentés dans les différents diagrammes de classes précédents, l'**IBD** permet de décrire la collaboration et les interactions inter-holons et inter-systèmes. Nous représentons un établissement de santé comme un **HMS** divisé en deux systèmes distincts : un **SIH** et un atelier. L'atelier correspond au **CDM**. Nous considérons que les logiciels qui ont des applications sur le **CDM** font partie du **SIH** et du **CDM**. Certains logiciels peuvent servir aussi dans le **HMS** et donc composer la partie informationnelle d'un holon. Dans [Huet 09c], nous avons vu le **SIH** comme un **SdS** [Boardman 06], c'est-à-dire avec plusieurs systèmes interopérants entre eux.

6.4.2.2 Représentation d'une dispensation globale et globalisée

Nous pouvons alors instancier les méta-modèles du niveau M2 des figures 5.6 et 5.7 de la section 5.5. L'**IBD** de la figure 6.11 représente un fonctionnement du **CDM** avec une dispensation globalisée.

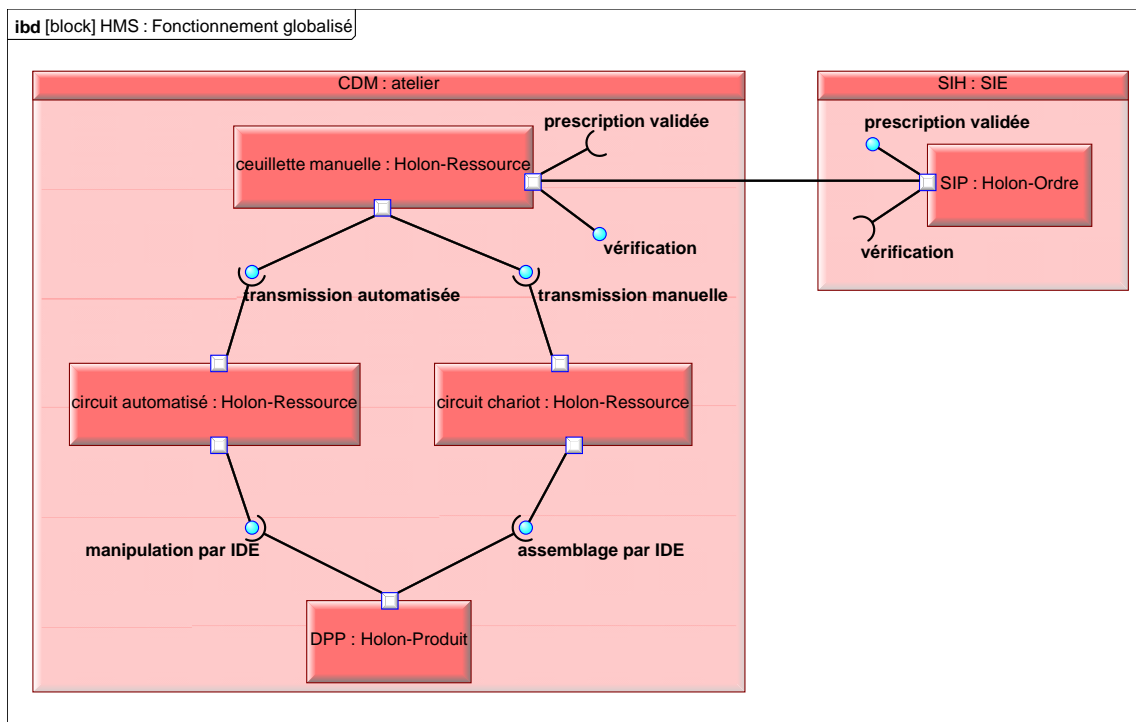


FIGURE 6.11 – ED pour un fonctionnement globalisé

Cette représentation serait adaptable à une dispensation globale en considérant la prescription envoyée par le **HO** comme la demande de médicaments venant d'un service. La cueillette manuelle correspond à une phase de création des *Doses Préparées pour un Service* (**DPS**), qui sont ensuite transmises au service via un circuit automatisé ou un circuit chariot. Les *Infirmiers Diplômés d'État* **IDE** préparent et administrent alors les doses pour chaque patient. La préparation des doses pour chaque patient n'est effectuée que dans les unités de soins sans relations avec le **SIH** et avec interventions humaines des **IDE** (risque d'erreurs).

6.4.2.3 Représentation d'une dispensation globale et globalisée

Dans le diagramme de la figure 6.12, les **DPP** communiquent avec toutes les ressources utilisées et doivent interopérer avec elles.

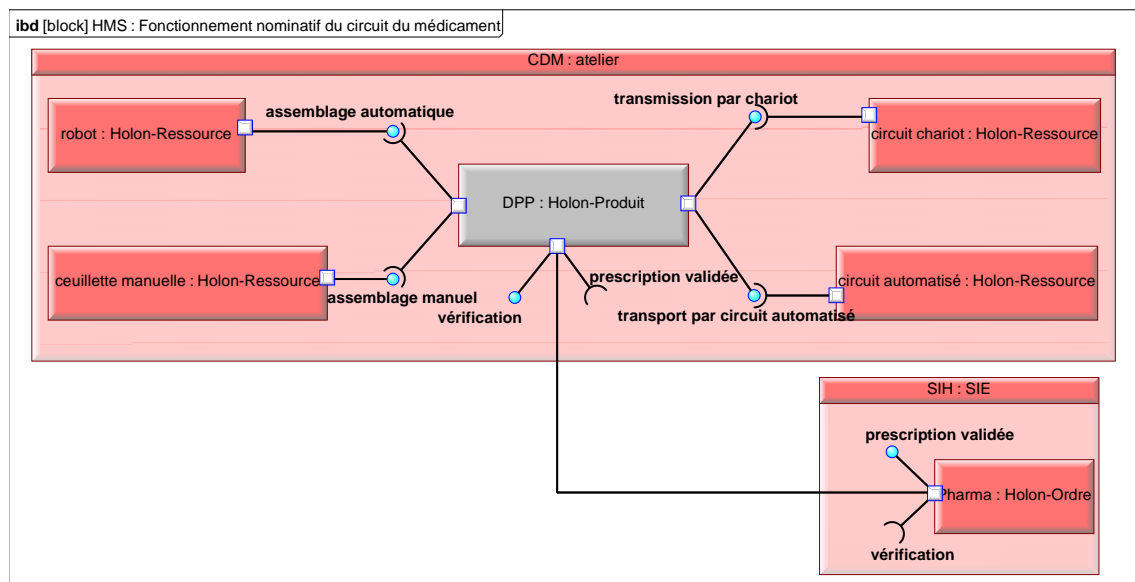


FIGURE 6.12 – ED pour un fonctionnement nominatif

Les **DPP** sont mises au cœur du système. Le produit, à l'aide de nouvelles technologies comme la **RFID** ou le code-barre, peut communiquer directement avec le **SIH** et la partie informationnelle des autres holons. Nous rejoignons ici le concept de « Système Contrôlé par le Produit ». Nous pouvons nous appuyer sur les travaux effectués actuellement sur ce domaine (cf chapitre 2) pour sécuriser et tracer le produit. C'est dans ce contexte que nous utilisons le paradigme de « interopérabilité contrôlé par le produit » que nous avons présenté au chapitre 2. Les **DPP** interopèrent avec le logiciel qui gère le circuit chariot, le circuit automatisé, le robot ou la cueillette manuelle. Le même logiciel peut être utilisé par plusieurs holons pour récupérer des données différentes.

Ces diagrammes correspondent au modèle du niveau M1 de la méthodologie **MRHMS**, c'est-à-dire que ces deux modèles sont une instantiation et sont conformes aux méta-modèles représentés par les figures 5.6 et 5.7. Nous voyons chaque holon comme un système interopérant avec le **SIH** ou d'autres holons (les holons peuvent être vus comme des systèmes indépendants), chaque interaction ayant un équivalent physique et informationnel. Par exemple, le transport du produit doit être choisi par l'intelligence du produit lui-même (partie informationnelle) et cette décision va provoquer la prise en charge de ce produit par le circuit concerné (partie physique). L'approche par composant nous sert à modéliser un **SdS**. Un **CDM** peut être vu comme un composant ayant des interactions avec un **SIH**. Dans l'atelier, chaque holon peut être vu comme un système ou composant, ayant lui-même des interactions avec d'autres holons. Nous voyons donc chaque composant comme un système, et chaque décomposition de composant comme un « accès » aux systèmes de niveau d'abstraction inférieur. Nous avons représenté sur ces deux diagrammes les interactions que nous pouvons contrôler. Par exemple, les interactions entre le robot et le **SIH** existent, mais elles dépendent du type de robot et de logiciel du **SIH**. Ces diagrammes représentent la boîte blanche du **CDM**. Nous pouvons

seulement voir ici les boîtes noires pour chaque holon. Une décomposition de chaque holon est effectuée avec un exemple de scénario dans le prochain chapitre.

6.5 PHASE 1 - BILAN

6.5.1 La démarche de réingénierie d'un circuit du médicament

L'intérêt de notre démarche est de pouvoir faciliter la réingénierie pour le contrôle par le produit de systèmes manufacturiers. Son enjeu est d'importance pour augmenter la flexibilité et la modularité de ce type de système. Dans l'exemple étudié ici, son intérêt est l'aide à la réingénierie d'un **CDM** pour permettre son redimensionnement, et ainsi pouvoir analyser et identifier les différents éléments à modifier. La réingénierie du **CDM** consiste en une réorganisation pour passer à un circuit plus sécurisé avec une dispensation nominative qui peut être apparentée à un *Système Contrôlé par le Produit* (**SCP**).

Les étapes d'analyse et de spécification de **ASCI-RHMS** de cette phase 1 nous ont permis d'isoler les composants qui seront instanciés sur le modèle du niveau M2. Cette instanciation, qui correspond à la « transformation 2 », nous donnera les *Entités Décisionnelles* (**ED**) écrites en **SysML**. Cette décomposition systémique ainsi que les diagrammes des **ED** nous ont fourni le *Modèle Générique de Connaissance* (**MGC**) du **CDM**.

Dans la phase 2, la méthodologie **ASCI** « académique » préconise d'analyser et de spécifier les entités décrites dans le **MGC**, afin d'obtenir des composants logiciels qui seront stockés dans une bibliothèque. Nous avons amélioré **ASCI** de deux manières :

- D'une part, l'intégration de **MBCSA** dans notre approche permet d'ajouter à cette bibliothèque le concept d'éléments ayant une partie physique, tels les holons, dans le cadre de systèmes automatisés ;
- Nous profitons de cette étape pour faire également la transformation d'un système contrôlé par les ressources vers un système contrôlé par le produit, comme nous le verrons plus en détail dans cette section.

Pour réaliser ceci, nous utilisons le langage **SysML**, tel que préconisé par **MBCSA**, qui nous permet de formaliser facilement l'approche par composant. En effet, le langage **SysML** est adapté à notre propos grâce à la notion de « port » et d'« interface » et à la décomposition par niveau d'abstraction qu'il permet. Chaque *Internal Block Diagram - diagramme interne de bloc* (**IBD**) de **SysML** peut se décomposer en autres **IBD** et à chaque niveau d'abstraction, on peut assembler des composants dans des contextes différents avec un **IBD** par contexte. On peut alors étudier **différents scénarios** ou **différentes implémentations** sur **différentes plateformes** de façon cohérente.

Le langage **SysML** nous propose donc un ensemble d'outils qui permettent une étude complète de n'importe quel holon. Il nous permet de modéliser un ensemble d'entités décisionnelles autonomes ainsi que leurs interactions, c'est-à-dire tous les éléments composant un **SCP**. Pour faire une transformation d'un système contrôlé par les ressources vers un système contrôlé par le produit, pour chaque composant que nous avons isolé dans la phase 1 (**DPP**, robots, armoire sécurisée, circuit de transport...), il faut choisir le diagramme le plus adapté, repérer les changements d'interaction et enfin le modéliser (cf. Phase 2 et Bilan phase 2). Le même processus peut être effectué pour les interfaces entre composants. La phase 1 a permis de décrire les éléments et la structure générale d'un **CDM**, la phase 2, présentée dans les sections ci-dessous, montre des exemples de ces processus de décomposition.

6.5.2 Création d'une bibliothèque de composants

Après les phases d'Analyse et de Spécification, la méthodologie **ASCI-RHMS** préconise l'utilisation des phases de Conception et d'Implémentation qui permettront la création d'une bibliothèque de composants réutilisables.

L'étape de *Conception* de la phase 2 pour **ASCI-CDM** va consister en la décomposition de chaque composant présenté à la figure 6.11 et 6.12 en boîte blanche. Cela signifie une nouvelle phase d'Analyse et de Spécification pour chaque composant afin de définir sa structure et son comportement interne. Dans cette phase, chaque nouvelle Analyse et Spécification correspondra à cet acte d'« accès » au niveau inférieur ; la boîte blanche d'un niveau d'abstraction correspondant à la représentation en boîtes noires de plusieurs autres composants constituant cette boîte blanche.

L'étape d'*Implémentation* de la phase 2, qui va suivre, correspondra au codage dans différents langages (langage de programmation, de simulation, automate,...) de chacun de ces composants. Ces deux phases permettront la création d'une bibliothèque de composants qui pourront être réutilisés dans des situations différentes lors de l'instanciation à un système du domaine (niveau Mo).

6.6 PHASE 2 - CONCEPTION ET IMPLÉMENTATION : CAS D'ÉTUDE D'UNE INTERFACE DE LOGICIEL DE SIMULATION

6.6.1 Conception

Nous avons développé une interface d'entrée/sortie compatible pour plusieurs logiciels de simulation (Automod et Witness). Dans cette étape, il s'agit de modéliser la structure et le comportement interne des composants. Cette interface permet de rentrer tous les paramètres que nécessitent les logiciels de simulations, elle permet de générer les fichiers texte qui seront en entrée des logiciels, ainsi que de récupérer les données envoyées par ces logiciels à la fin d'une itération de la simulation. Pour illustrer la façon dont nous avons développé cette interface, nous proposons de nous concentrer sur un composant : le circuit chariot. Le circuit chariot est un **HR** du **CDM**. Il est utilisé pour transporter les doses préparées de la pharmacie à l'unité de soins. Une livraison des doses par chariot demande l'utilisation de ressources humaines, c'est pour cette raison que généralement la livraison d'un service par chariot n'est effectuée qu'une fois par jour au maximum. Pour nous, le circuit chariot est modélisé par un **HR**, c'est-à-dire une des ressources qui peut être utilisée par le **HP** (Les **DPP**). Sa partie physique est le chariot lui-même, et sa partie informationnelle est l'agent logistique qui utilisera le **SI** correspondant à la gestion du circuit chariot. L'utilisateur doit rentrer tous les caractéristiques du chariot sur l'interface et leur nombre est directement calculé.

Un exemple de diagramme utilisé pour la création de cette interface est le diagramme d'activité technique de la figure 6.13 qui permet ce calcul. Il dépend des données reçues dans le modèle d'entrée. La liste des médicaments à envoyer pour chaque unité de soins peut être donnée en nombre de colis ou en nombre de lignes.

Un chariot peut être soit rempli avec des colis (par exemple, la figure *a* de la table 6.2) soit directement avec des médicaments. Dans ce cas, il existe deux types de chariots, soit un chariot de transport pour aller de la pharmacie au service (par exemple, figure *b*), soit un chariot de soin (par exemple, figure *c*) qui peut être équipé de matériel informatique.

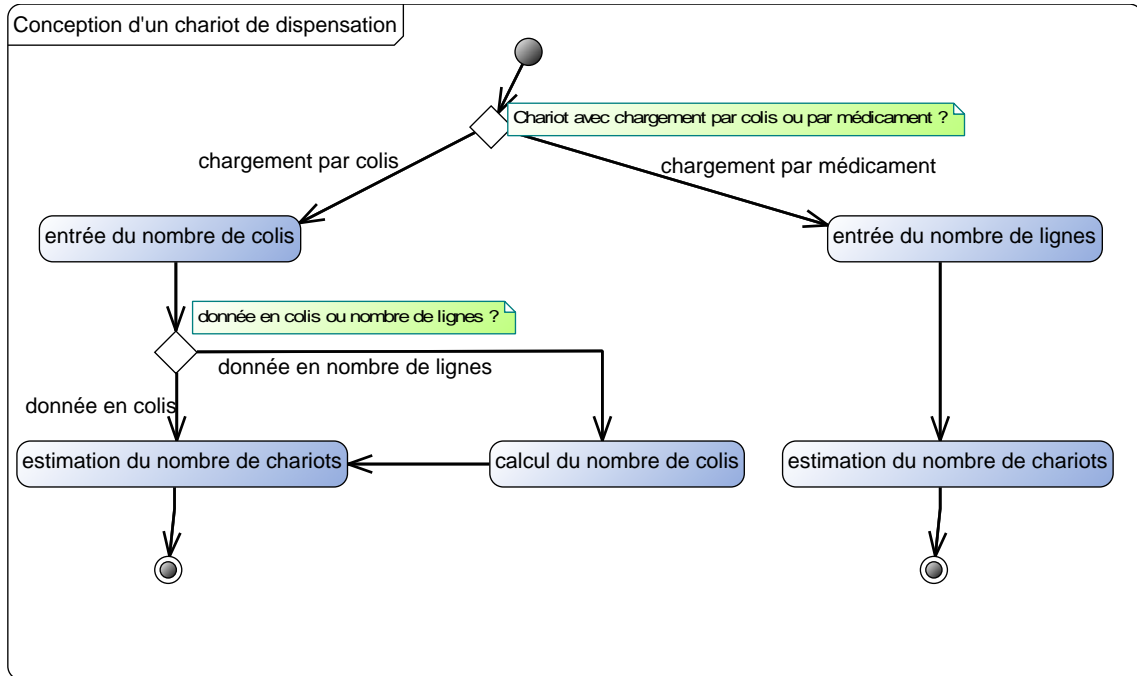


FIGURE 6.13 – Diagramme d'activité technique pour la conception d'un chariot



a) Chariot à colis



b) Chariot de transport à tiroirs



c) Chariot de soin à tiroirs

TABLE 6.2 – Différents types de chariot

Les chariots *b* et *c* sont des chariots de la société Rubbermaid¹. Ce diagramme permet de trouver le nombre de chariot dont nous avons besoins suivant sa capacité.

6.6.2 Implémentation

La phase d'Implémentation est le codage à partir des diagrammes d'activités techniques réalisés lors de l'étape de conception. Le travail de codage a été réalisé avec un étudiant en licence logistique qui l'a présenté dans son rapport de stage [Gayraud 08].

Reprenons notre exemple de chariot. Chaque chariot peut être utilisé pour tester différentes organisations qui utilisent différents types de chariots. Nous implémentons ce composant sur une interface d'entrée écrite en VB et nous le créons pour le logiciel de simulation Automod et Witness. La même méthodologie est appliquée pour chaque composant de notre système avec différents diagrammes et d'autres phases d'Analyse et Spécification quand cela est nécessaire : nous utilisons toujours des outils écrits en SysML, soit le IBD ou le BDD pour les composants qui sont composés par d'autres composants, soit des diagrammes d'activité technique pour décrire leurs comportements internes. La figure 6.14 montre une interface dans Excel qui permet l'utilisation de ces différents diagrammes tel que le diagramme d'activité technique de la figure 6.13 à partir des données entrées par l'utilisateur.

A partir des données entrées (soit en nombre de lignes, soit en nombre de colis) et de l'hypothèse de fonctionnement, cette interface calcule automatiquement le nombre de contenants (chariots).

The screenshot shows an Excel spreadsheet with the following data:

Paramètres			
Nombre d'éléments par cartouche	3		
Monte charge			
Nombre de contenant par monte-char	4		
Temps de passage entre 2 étages	0,2		
Hypothèse de livraison			
Hypothèse de livraison	1		
Les agents logistiques effectuent les tournées en entier	1		
Les agents logistiques déposent les rolls dans chaque secteur	2		
Hypothèse de fonctionnement contenant 1			
Hypothèse de fonctionnement contenant 1	2		
Hypothèse de fonctionnement contenant 2			
Hypothèse de fonctionnement contenant 2	2		
Hypothèse de fonctionnement contenant 3			
Hypothèse de fonctionnement contenant 3	2		
Fonctionnement par ligne	1		
Fonctionnement par colis	2		
Hypothèse de tournée			
Hypothèse de tournée	1		
Donnée d'entrée			
Donnée d'entrée	2		
Ligne	1		
Colis	2		
Rangement			
	C1	C2	C3
Temps de rangement médicamenteux	0	0	0
Temps de rangement soluté	0	0	0
Temps de rangement DMS	0	0	0
Temps de rangement gaz	0	0	0

On the left side, there are three sections for containers:

- Contenant 1:**
 - Nombre de médicament: 187
 - Nombre de coluté: 0
 - Nombre de DMS: 0
 - Nombre de gaz: 0
 - Nombre de contenants: 1
- Contenant 2:**
 - Nombre de médicament: 0
 - Nombre de coluté: 30
 - Nombre de DMS: 0
 - Nombre de gaz: 0
 - Nombre de contenants: 1
- Contenant 3:**
 - Nombre de médicament: 300
 - Nombre de coluté: 100
 - Nombre de DMS: 0
 - Nombre de gaz: 0
 - Nombre de contenants: 1

FIGURE 6.14 – Interface utilisateur Excel de l'interface entrée/sortie de logiciels de simulation

1. <http://www.rubbermaidmedical.com> (accessible au 4 octobre 2010)

6.7 PHASE 2 - CONCEPTION ET IMPLÉMENTATION : CAS D'ÉTUDE DU SUIVI DU CIRCUIT DU MÉDICAMENT PAR IDENTIFICATION RADIOFRÉQUENCE (RFID)

6.7.1 Cahier des charges

L'objectif principal de ce projet est de sécuriser le **CDM**, c'est-à-dire de fournir une aide à l'administration du médicament, puis de rendre compte de l'administration du médicament. Pour cela, les pharmaciens prévoient la mise en place d'une dispensation nominative au niveau de la pharmacie. Ce travail a été réalisé avec une étudiante en dernière année d'école d'ingénieur que nous avons encadré et qui est présenté dans son mémoire de projet de fin d'études [Vogel 08].

Pour cette étude, nous avons suivi une démarche préconisée par plusieurs méthodologies pour la conception de systèmes multi-agents comme **MaSE** (cf section 3.3.4.1) ou **TROPOS** (cf section 3.3.4.2), ainsi que par [Kisti 05] pour la conception de composants d'IHM (Interface Homme-Machine) modulaire pour les systèmes de production. Elle consiste à décrire le contexte de l'étude, puis à créer un diagramme de cas d'utilisation qui permettra de détailler le rôle de chaque acteur du système afin de mieux le concevoir.

Au niveau de l'**IDE**, nous prévoyons une vérification automatisée des éléments suivants au moment de l'administration de médicaments :

- le bon médicament,
- au bon patient,
- la bonne dose,
- le bon horaire,
- la bonne forme.

La solution que nous souhaitons mettre en avant est une identification du sachet contenant les doses à administrer pour une tranche horaire donnée afin de permettre cette vérification. La dispensation ne serait pas journalière mais échelonnée en fonction des horaires d'administration du médicament et en fonction de la visite du médecin, et nominative. Dans ce cas, la livraison s'effectuera par un circuit automatisé car ce système de transport est plus flexible. En effet, une livraison des doses par chariot demande l'utilisation de ressources humaines, c'est pour cette raison que généralement la livraison d'un service par chariot n'est effectuée qu'une fois par jour au maximum.

Nous mettons en place un système de lecture/écriture des étiquettes nominatives de prescription. Dans le cas de livraisons particulières (volume ou poids trop importants pour un circuit automatisé, frigorifiés, fragiles, etc. . .), nous avons décidé, avec les pharmaciens, d'une solution alternative au circuit automatisé consistant en l'utilisation d'un chariot à tiroirs pour une livraison dans les services effectuée par un agent logistique (comme sur la figure *b*) de la table 6.2).

Voici les éléments pris en compte pour la conception de ce logiciel de suivi :

1. prescription par le médecin avec informatisation de cette prescription,
2. dispensation par le pharmacien comprenant l'analyse pharmaceutique de l'ordonnance, la validation de la prescription, la délivrance et la livraison vers les différents services de l'hôpital,
3. administration par l'**IDE** et renseignement de cette administration au niveau du **SI** général.

Dans ce projet, nous étudions l'insertion dans les sachets de tags **RFID** compatibles lecture/écriture de capacité suffisante pour contenir les données de la prescription horaire avec une récupération au moment de l'administration pour réutilisation.

Un convoyeur avec des conteneurs est utilisé pour récolter les éléments sortant des robots de préparation de prescription. Un tag **RFID** est programmé et ajouté au contenu. Cette opération doit avoir lieu entre le début du remplissage des conteneurs et son arrivée à l'ensacheuse.

Le contenu, c'est-à-dire les médicaments et le tag **RFID**, est renversé dans l'ensacheuse. Plusieurs utilisations du tag peuvent être envisagées :

- Aide au tri en sortie d'ensacheuse : pour éviter d'avoir à communiquer les données de la prescription à l'ensacheuse, on pourrait envisager une donnée supplémentaire dans le tag synthétisant les conditions de passage du sachet dans le circuit automatisé ou par chariot comme par exemple, un calcul du poids et du volume, la reconnaissance de la nature des médicaments et des conditions de livraison. Ces calculs doivent être effectués en amont de l'antenne **RFID**.
- Aide au rangement des sachets par l'**IDE** : une fois arrivé dans le service, le sachet est pris en charge par l'**IDE**. Le tag **RFID** permet de ranger le sachet dans le tiroir correspondant au patient à traiter. Pour faire sa tournée, l'**IDE** utilise un chariot contenant au moins autant de tiroirs que de lits dont elle doit s'occuper, chaque tiroir étant associé à un lit, et équipé d'un lecteur **RFID**. La correspondance lit-patient doit être facilement consultable.
- Vérification des 5 règles au niveau de l'administration : la lecture du tag sachet et du tag patient permet de vérifier que le bon médicament, avec la bonne dose, la bonne forme, le bon horaire est administré au bon patient. De plus, le login ou la lecture du badge de l'**IDE** permet de savoir qui est responsable de l'acte. Une validation automatique de la délivrance des médicaments au patient est possible à ce moment.

Le but est de faire un dispositif d'écriture des tags au moment de la préparation des prescriptions dans la pharmacie, et de la lecture des tags par l'**IDE** au moment de l'administration des médicaments au patient. Il faut aussi que ce dispositif soit bien intégré dans le **SI** global.

Les objectifs sont donc l'écriture sur le tag **RFID** du contenu de la prescription :

- les médicaments (identifiant),
- ID patient,
- horaire d'administration.

et la lecture des tags **RFID** pour l'administration au patient :

- bracelet patient,
- carte infirmière,
- étiquette sachet.

L'interface utilisateur doit comporter les deux volets correspondants. Lors de l'écriture, il faut que l'on puisse voir les données entrées dans la mémoire du tag à l'écran. Lors de la lecture, on doit pouvoir visualiser sur l'écran :

- l'identité du patient,
- l'identité du soignant,
- la liste des médicaments avec nom, dose, forme,
- l'horaire d'administration,
- la validation si tout est conforme après comparaison des données.

Le composant ainsi créé (le dispositif d'écriture et de lecture) correspondra à l'interface « vérification » entre le **HP** et le **HO** du diagramme représentant le fonctionnement en nominative de la figure 6.12. Il pourra être implanté par exemple sur les chariots de soin de la figure c) de la table 6.2.

6.7.2 Conception

A partir de ce cahier des charges, nous devons identifier les composants à modéliser et décrire leur fonctionnement, c'est-à-dire identifier les entités du **MGC** que nous utiliserons. Nous utilisons le diagramme représentant un fonctionnement avec une dispensation nominative (figure 6.12). Nous pouvons ainsi décomposer les composants rentrant en jeu. La partie informationnelle du **HP** sera la puce **RFID**, qui devra recevoir la prescription validée par le **SIH**. Ce logiciel, composé d'un dispositif de lecture et d'un dispositif d'écriture, fera alors le lien entre le logiciel d'écriture et de validation de prescription (le **HO**) et la puce **RFID**. Nous avons représenté le lien et le rôle de chacun de ces composants du **MGC** dans des diagrammes de cas d'utilisation d'**UML** et de **SysML**. Les diagrammes de cas d'utilisation se composent d'acteurs (représentés par des silhouettes) et de cas d'utilisation (représentés par des ellipses). Ces diagrammes montrent les relations qui existent entre des acteurs et des fonctionnalités du système.

Ce logiciel devra faire partie du Programme Directeur de Production (l'aiguilleur) et devra s'appuyer sur les décisions centralisées prises par les **DPP**. Nous avons détaillé le modèle correspondant avec quatre diagrammes :

- la figure 6.15 représente l'identification des tags et leur programmation,
- la figure 6.16 représente le rôle des acteurs dans le choix du mode de transport dans la pharmacie,
- la figure 6.17 représente ce rôle lors d'une distribution des **DPP**,
- la figure 6.18 le représente lors de l'administration au patient.

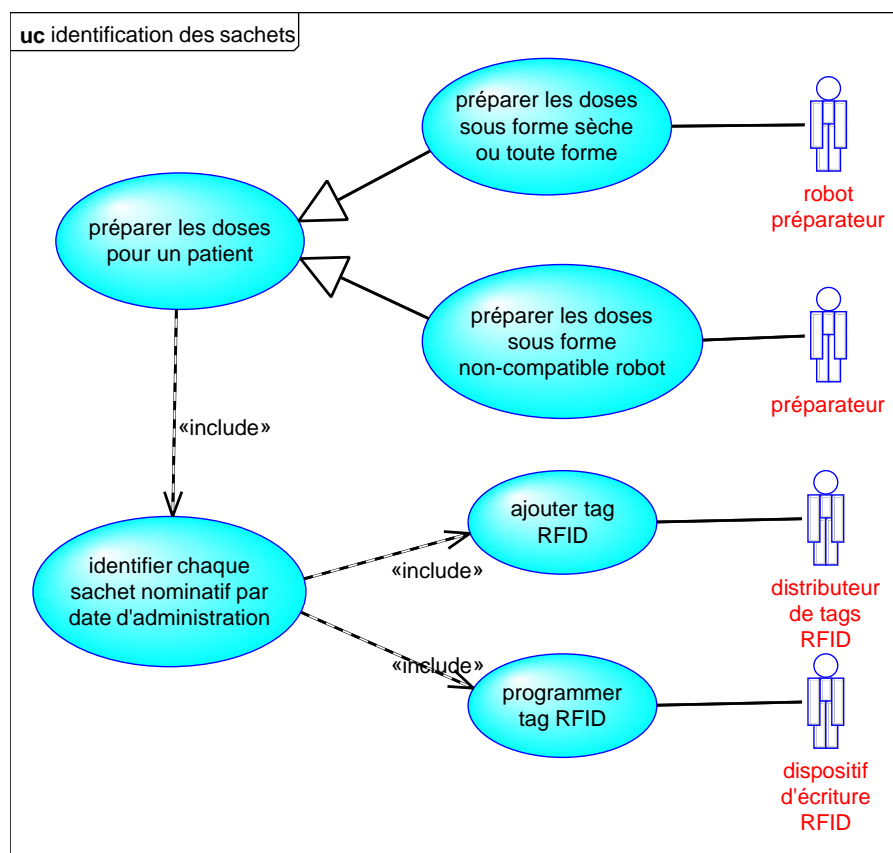


FIGURE 6.15 – Identification des sachets dans la pharmacie

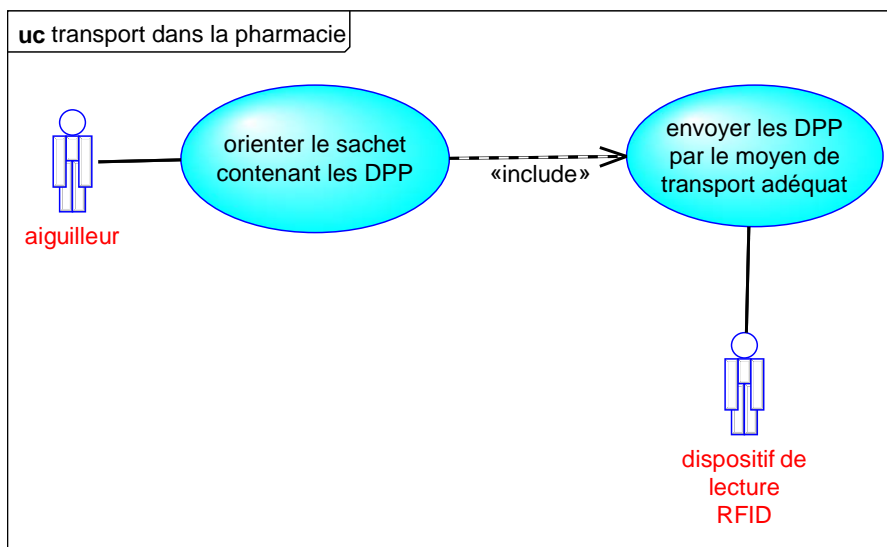


FIGURE 6.16 – Choix du moyen de transport dans la pharmacie

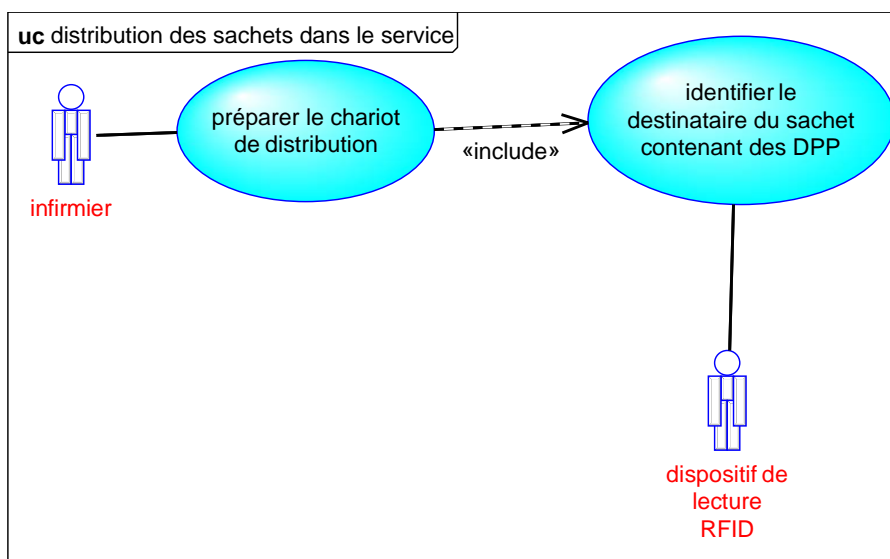


FIGURE 6.17 – Préparation de la distribution des sachets dans un service

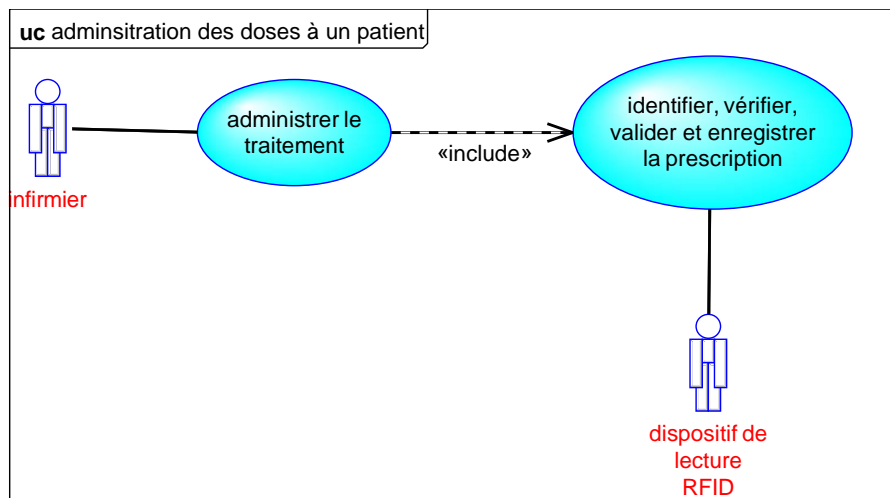


FIGURE 6.18 – Administration des médicaments au patient

Les acteurs représentés sont des composants et des classes du **MGC** construits dans la phase 1. Une fois ces rôles définis et le lien entre chaque composant et classe explicité, nous pouvons passer au codage du logiciel.

6.7.3 Implémentation

6.7.3.1 Choix des logiciels de programmation

Le langage de programmation utilisé est le C# et le développement est fait sous MS Visual Studio. La démarche de programmation qui a été suivie est découpée en une partie « recherche dans base de données » et une partie « gestion des tags RFID ». La base de données est gérée dans MS ACCESS. On peut voir le modèle relationnel sur la figure 6.19.

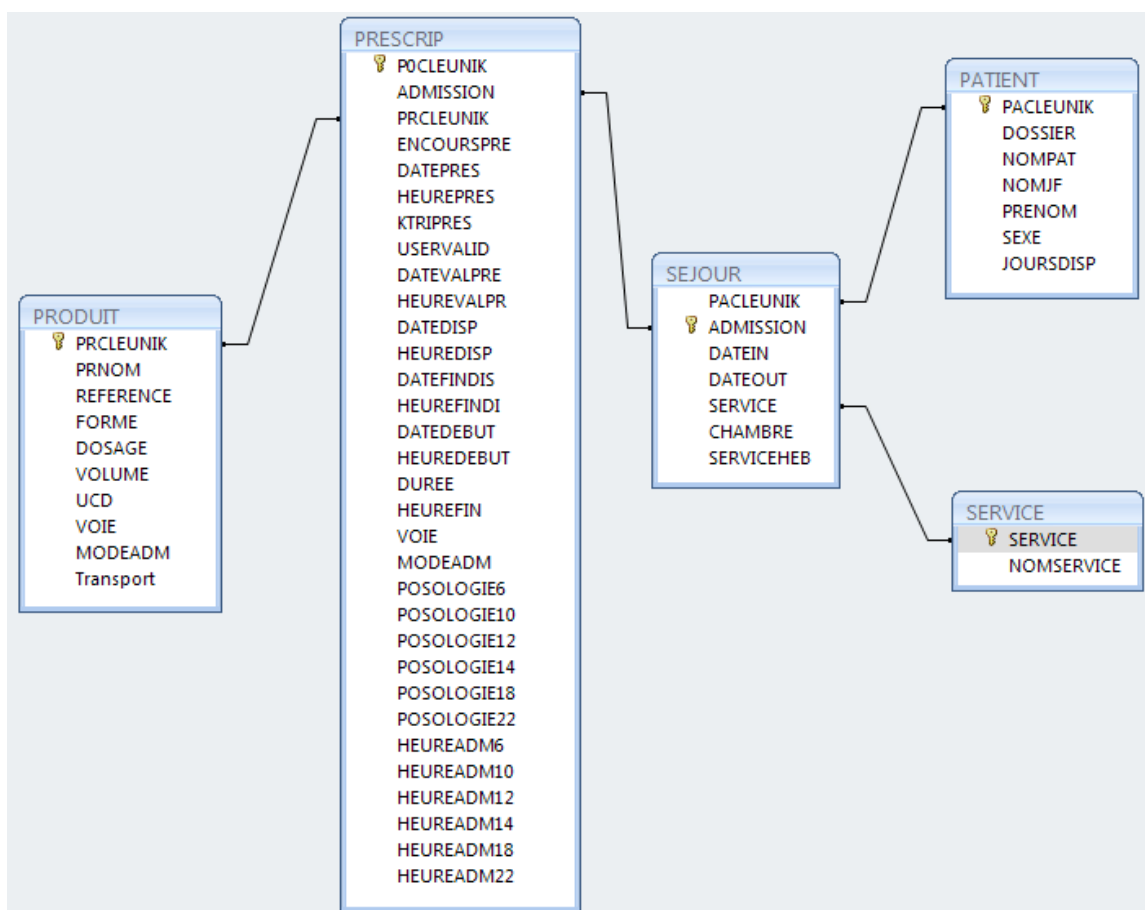


FIGURE 6.19 – Diagramme relationnel de la base de données géré par le dispositif

Ce modèle relationnel détaille toutes les tables dont nous avons besoin pour pouvoir faire fonctionner ce dispositif d'écriture et de lecture de DPP. Les éléments suivants de la base de donnée sont affichés à l'écran pour tous les patients qui sont concernés par la dispensation au regard de leurs prescriptions pour la prochaine tournée de soin :

- PATIENT.NOMPAT = nom du patient
- PATIENT.PACLEUNIK = identifiant patient
- PRESCRIP.POSOLOGIE(numeroadmin) = posologie correspondant à l'horaire d'administration pour un médicament donné
- PRESCRIP.VOIE = voie
- PRESCRIP.DUREE = durée d'administration (exemple : cas d'une perfusion)
- PRESCRIP.KTRIPRES = désignation du médicament, dosage, forme
- PRODUIT.UCD = Unité Commune de Dispensation

- SEJOUR.SERVICEHEB = service d'hébergement
- SEJOUR.CHAMBRE = numéro de chambre
- PRODUIT.Transport = mode de transport

Des détails techniques sur la programmation du tag et de l'interface sont donnés dans l'annexe A.2.

6.7.3.2 Préparation

A l'affichage de l'interface, on trouve : en haut à gauche, l'horaire de la prochaine administration, qui correspond plus précisément à la prochaine tournée de soin par les IDE du service ; en haut à droite, l'heure et la date courantes. Un exemple d'affichage après l'écriture d'un tag est montré à la figure 6.20.

Nom Patient	ID	Posologie	Voie	Durée	Produit
FAUXNOM	213165	1	54	0	CORDARONE 200 mg, cpr
FAUXNOM	213165	0,5	54	0	LASILIX 40 mg, cpr
FAUXNOM	213165	2	54	0	ZYLORIC 100 mg, cpr
FAUXNOM	213165	1	54	0	CASODEX 50MG CP
FAUXNOM	213165	1	54	0	REVATIO 20 mg, cpr
AUBIERE	217888	1	49	1	SODIUM BICARBONATE 1,4% LAVOISIER, flac 500 m
AUBIERE	217888	1	54	0	PLAVIX 75 mg, cpr
AUBIERE	217888	1	54	0	KARDEGIC 75 mg, pdr pr sol buv, sachet
AUBIERE	217888	1	62	0	ACTRAPID 100 lu/mL, sol inj, cart 3 mL PENFILL
DUPOND	231554	1	106	9	ATROVENT, susp pr inhal, flac 10 mL
DUPOND	231554	1	54	9	DIFFU-K 600 mg, gélule
DUPOND	231554	3	54	2	DIFFU-K 600 mg, gélule
DUPOND	231554	1	54	0	KARDEGIC 75 mg, pdr pr sol buv, sachet
CLERMONT	242922	1	54	1	LOXEN LP 50 mg, gélule
MARTIN	303524	1	54	0	PLAVIX 75 mg, cpr
MARTIN	303524	1	54	0	KARDEGIC 75 mg, pdr pr sol buv, sachet
BEAUMONT	304769	1	54	0	PLAVIX 75 mg, cpr
BEAUMONT	304769	1	54	0	KARDEGIC 75 mg, pdr pr sol buv, sachet
DUPONT	324243	0,5	54	3	URBANYL 10 mg, cpr
DUPONT	324243	1	49	19	ZOPHREN 4 mg/2 mL, sol inj, amp
DUPONT	324243	1	54	3	NEURONTIN 600 mg, cpr

FIGURE 6.20 – Exemple d'affichage après l'écriture du tag à la pharmacie

Le nombre de patients à traiter pour cet horaire s'affiche en bas de la page au-dessus des boutons. La partie de gauche comporte les données provenant directement de la table issue du SIP (figure 6.19). Chaque ligne du tableau correspond à une ligne de prescription. La requête SQL qui le génère est dans le code du formulaire « Préparation ». La liste déroulante en haut à droite contient tous les noms des patients à traiter. Pour chacun d'entre eux, une étiquette est générée et affichée à droite : c'est l'image de ce que contiendra l'étiquette RFID.

Voici la description du menu :

- RAFRAICHIR : réaffiche la page et change l'heure courante, donc peut changer l'horaire de la prochaine administration. Après un rafraichissement de la page, il faut resélectionner un patient dans la liste déroulante,

- EFFACEMENT TAG : écrit du vide sur le tag, si l'opération a réussi, affichage de « TAG EFFACE » sur le bouton,
- LECTURE TAG : lit les données et les affiche dans la fenêtre « étiquette » à la place des données de la base,
- ECRITURE TAG : écrit sur le tag les données provenant de la base de données sur des emplacements prédéfinis sur le tag (paramètres dans un fichier au format *dll*). Si l'opération a réussi, affichage de « ECRITURE OK » sur le bouton,
- VERIFICATION : effectue une vérification des données du tag avec celles de la base de données. Si l'opération a réussi, affichage de « VERIF OK » sur le bouton,
- ECRITURE + VERIFICATION : combinaison des 3 fonctions précédentes. Si l'opération a réussi, affichage de « ECRITURE OK VERIF OK » sur le bouton,
- MENU : retour au menu principal.

Si une erreur est survenue, que ce soit pendant l'effacement, l'écriture ou la vérification, un message s'affiche en précisant la nature du problème.

6.7.3.3 Administration

A l'affichage de la page, on retrouve en haut à gauche l'horaire de la prochaine administration, et en haut à droite l'heure et la date courantes. Tous les autres champs du formulaire sont vides. Un exemple d'affichage après lecture du tag est montré sur la figure 6.21.

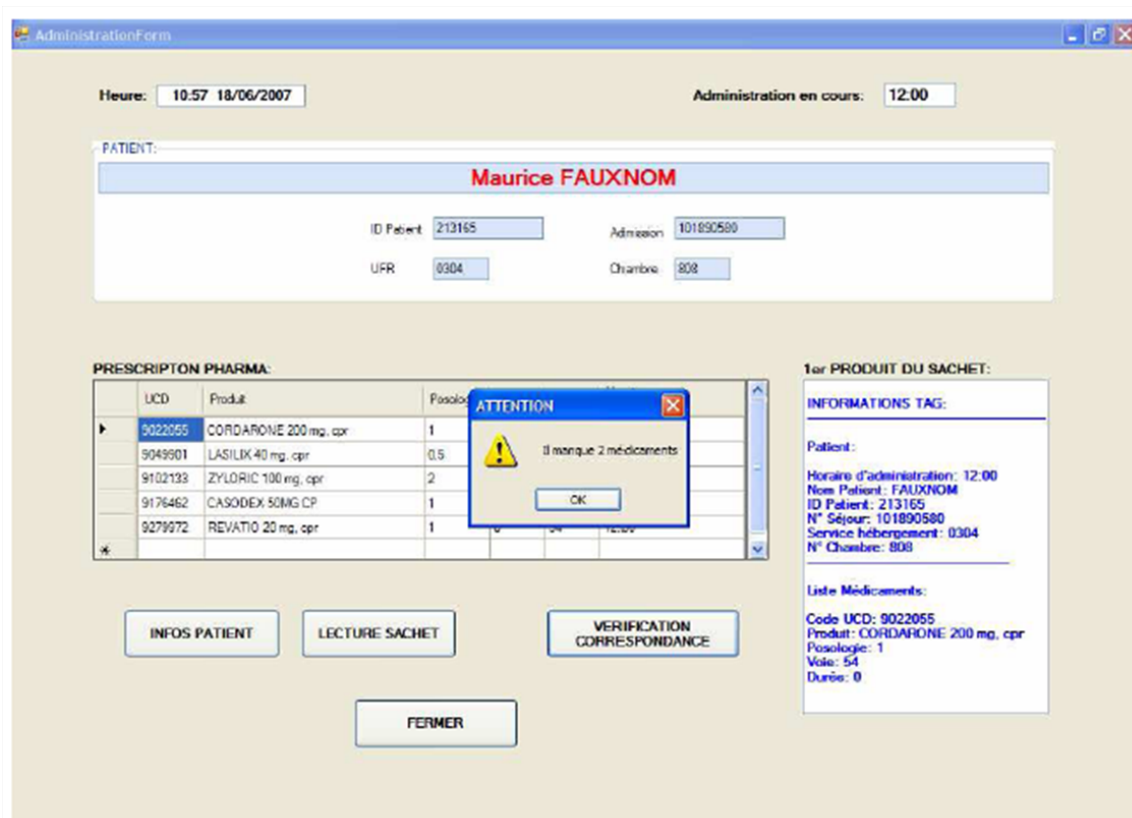


FIGURE 6.21 – Exemple d'affichage après la lecture pour la validation au lit du malade

Voici la description du menu :

- INFOS PATIENT : elle engendre deux actes. Partie Patient : lit les données du tag (Nom, Prénom, ID, N° Admission) et certaines données de la base de données (Service, Chambre). Partie Prescription : affiche les données de la base de données de la prescription en même temps que les données patient,
- LECTURE SACHET : affiche dans la fenêtre de droite les données du tag (qui ont été écrites à la pharmacie pendant la préparation des doses),
- VERIFICATION : compare les données du tag avec les données de la base de données en temps réel. Si tout correspond, le bouton devient vert et affiche : « DELIVRANCE VALIDEE ».

S'il y a une erreur de correspondance, le message d'erreur correspondant s'affiche. Dans l'exemple de la figure 6.21, le sachet ne contient que 6 médicaments différents alors que la prescription indique que le patient doit en recevoir 8.

Pour toutes les erreurs qui sont recherchées dans la fonction de vérification, il existe un message d'erreur similaire à celui-ci. Les actions de l'IDE restent à déterminer mais il est informé dans tous les cas de la non-correspondance entre ce qu'il s'apprête à administrer au patient et la prescription sous forme de base de données, dont la mise à jour sera faite, en théorie, quasiment en temps réel. Le code de la vérification au lit du malade est présenté dans l'annexe A.2.

6.8 PHASE 2 - CONCEPTION ET IMPLÉMENTATION : CAS D'ÉTUDE DE L'UTILISATION DU LOGICIEL DE SIMULATION AUTOMOD

6.8.1 État de l'art sur la simulation du circuit du médicament

Nous avons développé des modèles de simulation d'une pharmacie générique. Ce travail a été réalisé avec un étudiant en avant-dernière année d'école d'ingénieur que nous avons encadré [Viry 10].

Différents auteurs ont déjà cherché à simuler le CDM de différents hôpitaux dans plusieurs pays, dans le but d'évaluer les performances d'une configuration et de créer un outil d'aide à la décision. Nous présentons ici les travaux des auteurs qui nous paraissent les plus intéressants.

[Spry 05] a modélisé la pharmacie du BroMenn Regional Medical Center (BMRC), un hôpital de 224 lits localisé dans l'Illinois, aux USA. Dans cet article, les auteurs modélisent sous Automod son fonctionnement actuel pour optimiser le temps des ressources, en détaillant au maximum le temps de chaque acteur (jusqu'à mettre aussi dans le modèle le temps passé au téléphone). Le personnel pharmaceutique prépare les prescriptions pour chaque patient à la pharmacie. Un stock géré par ce personnel est disponible dans chaque service (armoires de stockage automatisées). Ils testent différents types de scénarios comme par exemple différents moments de remplissage des armoires de stockage. Ils peuvent envoyer les prescriptions préparées par circuit pneumatique ou par chariot. Ils testent aussi différents plannings de travail afin d'optimiser le temps de travail de chaque ressource.

[Yurtkuran 08] a effectué une simulation du circuit du médicament sous ProModel d'un hôpital en Turquie (635 lits). La pharmacie fonctionne en dispensation nominative. Ils ont testé plusieurs scénarios avec des horaires de travail différents dans le but de minimiser le temps d'utilisation des ressources. La pharmacie est ouverte 24h/24h et ils préparent les doses unitaires (les DPP) pendant la nuit. La liste des prescriptions commandées pendant la journée est envoyée à minuit sur le SI utilisé par cet hôpital. Leur objectif est de minimiser le temps de

dispensation en optimisant le temps de travail des ressources principalement entre l'équilibre des équipes de nuit et de jour, en centrant le modèle sur l'amélioration du **SI** utilisé à la pharmacie.

[Wong 03] a fait deux modèles (« AS-IS » et « TO-BE ») pour quantifier l'impact de la mise en place d'un nouveau **SI**. Un des deux modèles correspond à l'analyse du système avant la mise en place de ce nouveau système électronique (« AS-IS »), et l'autre modèle est la simulation dans le cas de sa mise en place (« TO-BE »). Ce nouveau système (Computer Physician Order Entry - CPOE) rend possible un contrôle automatique des ordonnances en plus du fait de permettre une prescription informatisée. Il permet aussi aux pharmaciens d'analyser plus rapidement et facilement les prescriptions des médecins. Ils travaillent sur l'hôpital Sunnybrook and Women's College Health Sciences Centre (SWCHSC) localisé à Toronto dans l'Ontario, au Canada. Sunnybrook compte 458 lits et 26000 visites de patients par an. Ces modèles sont aussi effectués sous ProModel. Ce modèle suit la prescription électronique puis sa préparation. Les comparaisons sont effectuées sur les critères suivants : temps de travail moyen de chaque ressource, taux d'erreur dont certaines dues au temps de vérification, de préparation, de livraison et de dispensation.

6.8.2 Bilan et perspectives sur la simulation du CDM

Dans tous ces travaux, une des perspectives citée est de rendre les modèles adaptables à d'autres hôpitaux. Nous nous sommes efforcés de rendre notre modèle le plus réutilisable possible. Dans nos travaux, nous nous intéressons plus à la partie « technique » du **CDM** en modélisant différents types de fonctionnement et en appliquant seulement un temps de préparation moyen, c'est-à-dire que nous nous intéressons peu à la partie « pharmaceutique » qui prendrait en compte un temps de validation et un temps de préparation différents suivant les médicaments et les services. Notre modèle se concentre sur le produit en lui-même et ne s'intéresse pas à l'écriture de la prescription. Par conséquent, une des différences par rapport à ces modèles est que les prescriptions sont mises dans des fichiers texte en entrée de notre modèle. Nous nous situons dans la perspective de l'article de [Yurtkuran 08] : les prescriptions sont déjà présentes au moment du démarrage du modèle. Une des perspectives d'amélioration de nos modèles serait de faire entrer dans le modèle un stock de « commande » en attente, en prenant en compte un temps d'analyse du pharmacien ou de commande du médecin (des horaires de commande). L'utilisation des données et les temps utilisés par [Wong 03] pourraient fournir une perspective à notre modèle en le complétant par l'ajout de pharmaciens. Une des limitations de notre modèle est que nous ne gérons pas les priorités dans les prescriptions (ex : commande urgente à la pharmacie). Nous avons aussi repris l'idée de [Wong 03] en créant une pharmacie avec une livraison globale qui correspond à notre « AS-IS » et une pharmacie avec une dispensation nominative qui correspond à notre « TO-BE ».

Un composant-système que nous avons conçu est le composant Pharmacie. Ce composant est générique dans le sens où il peut s'adapter à beaucoup d'établissements de santé en France. La pharmacie est représentée dans le **MGC** par une unité de soins.

6.8.3 Le logiciel de simulation Automod

6.8.3.1 Présentation de Automod

Automod est développé par Brooks Automation, Inc. C'est un logiciel de simulation en 3D. Ce logiciel simule en particulier les flux de production. On trouve notamment comme exemple de projets réalisés sous Automod la modélisation de convoyeurs dans une usine, la

modélisation d'un carrefour routier, ou encore la modélisation de la bagagerie d'un terminal de l'aéroport de Toronto.

Ce logiciel permet de simuler les convoyeurs de manière approfondie en incluant de nombreuses spécifications qui permettent un rendu très réaliste, comme par exemple le taux d'occupation des convoyeurs, le nombre de charges passées, les temps de cycle des charges... On peut de plus ajouter des temps de pause dans le cycle. On programme le système dans un fichier source. Ce dernier est programmé dans le langage spécifique d'Automod se rapprochant du C. Grâce à la simulation, on peut prévoir le fonctionnement du système et ainsi le dimensionner ou le redimensionner.

Tous les systèmes de production comportent deux types d'éléments : les éléments temporaires, qui sont produits et enlevés du système (comme les produits qui se déplacent à travers le système), et les éléments permanents (comme des personnes, des machines et du matériel).

6.8.3.2 Resources

Dans Automod, les personnes ou les machines qui accomplissent une tâche sont appelées les *Resources*. Les machines peuvent être des robots ou des ascenseurs. Les *Resources* sont des entités que l'on peut occuper, ou saisir, pendant un certain temps. Cela nous permet d'obtenir des résultats relatifs aux taux d'occupation et aux horaires d'utilisation.

6.8.3.3 Load

Une *Load*(charge) correspond à une entité du système. Elle représente une quantité de produits ou de matériaux qui se déplacent à travers le système. Les *loads* sont des éléments temporaires que l'on peut créer et détruire durant une simulation. Dans notre cas, nous avons 2 types de *Loads* représentant des entités physiques : les solutés (L_Solutés) et les médicaments (L_Medicaments) ainsi que des *Loads* représentant des flux d'information (L_init).

6.8.3.4 Queue

Les *Queues* représentent un espace physique dans lequel les charges ou les véhicules peuvent attendre. Les entités à stocker sont mises dans des *containers* qui peuvent être eux-mêmes composés de *sous-containers*. Il s'agit d'un lieu où sont stockées les entités durant un temps d'attente ou définitivement (par exemple dans les armoires des services). Une charge (*Load*) ne quitte pas une *queue* avant qu'elle ne soit capable d'atteindre sa prochaine destination. Une *Queue* possède les caractéristiques suivantes :

- capacité : le nombre de *Loads* qu'elle admettra ;
- nombre de *Queues* : une *Queue* peut être subdivisée en plusieurs *Queues*. On indique alors en combien de *Queues* elle est divisée ;
- représentation graphique : la façon dont elle va être représentée.

6.8.3.5 Système de déplacement

On peut utiliser dans Automod les systèmes de déplacement suivants pour déplacer des charges d'un endroit à un autre : le *path mover systems* (système modélisant un véhicule se déplaçant le long d'une voie) et des *convoyors*. Un *vehicule* est une entité se déplaçant le long d'un chemin et pouvant transporter des *loads*.

6.8.3.6 Order list

Pour gérer les *Load*, notamment lorsqu'ils sont en attente dans des *Queues* ou lorsqu'on veut les déplacer, on utilise des *orders lists*. Une *order list* est une liste qui classe et ordonne des *Loads* et des *vehicules*. De plus, on peut temporiser une *Load* de la manière suivante :

- soit en utilisant l'action *Wait* pour la temporiser pendant un certain temps ;
- soit en utilisant l'action *Wait to be ordered* pour la temporiser jusqu'à ce qu'un autre *Load* ou *vehicule* lui ordonne de continuer. A ce moment, il est possible de déplacer en une fois toutes les charges contenues dans l'*order list*.

Quand on définit une *order list*, on considère les caractéristiques suivantes :

- *Loads* ou *vehicules* qui sont en attente d'être ordonnés ;
- *vehicules* ou *loads* qui vont ordonner les autres ;
- Nombre d'*orders lists* : on définit une gamme d'*orders lists*, on indique ici combien d'*orders lists* la composent.

6.8.4 Simulation d'une pharmacie générique

6.8.4.1 Pharmacie générique pour une livraison globale

Pour étudier les relations et les interactions dans une pharmacie générique avec une dispensation globale ou globalisée, c'est-à-dire une livraison globale, nous avons utilisé le diagramme des ED de la figure 6.11. Cette pharmacie est composée des éléments du MGC suivants :

- une station de travail avec un préparateur ou un magasinier,
- un stock de médicaments,
- un stock de *Doses Préparées pour un Service (DPS)* rangées par pôle.

Les *DPP* sont assemblées à partir de ces *DPS* envoyées dans le service. Les demandes de services composées des dotations et de l'addition des prescriptions sont soit calculées après validation pharmaceutique au niveau de la pharmacie dans le cas d'une dispensation globalisée, soit faites par les *IDE* dans le cas d'une dispensation globale non équipé du système Plein/Vide, soit automatiquement par le *SI* du système Plein/Vide. Le modèle est juste composé d'un stock d'entrée, de préparateurs et de magasiniers puis d'un stock de rangement par pôle puis par service. Les demandes de chaque service sont données en entrée. La figure 6.22 représente un exemple de ce modèle.

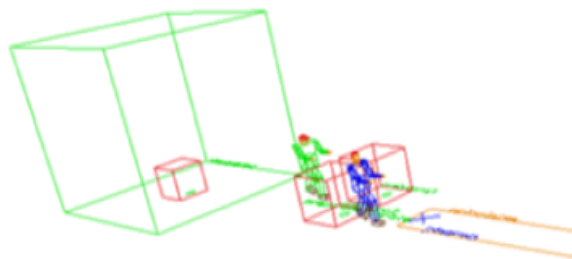


FIGURE 6.22 – Exemple d'une pharmacie en fonctionnement global du circuit du médicament

Cette figure est composée d'un stock de médicaments que les préparateurs ou magasiniers (en vert) utilisent, les *DPS* sont transmises ensuite sur le convoyeur via un agent logistique (en bleu) pour rejoindre leurs stocks.

6.8.4.2 Pharmacie générique pour une livraison nominative

Modéliser une pharmacie avec une dispensation nominative est plus complexe que dans le cas d'une dispensation globale et globalisée. Pour permettre d'isoler les entités du **MGC** dont nous aurons besoin pour la simulation d'une telle pharmacie, nous avons commencé par établir un cadre expérimental :

- **Objectif** : Simuler une pharmacie nominative suivant tous les cas possibles, et implantable facilement dans une simulation d'hôpital existante.
- **Variables d'entrée** :
 - Ordonnances (service/patient/médicament/posologie/horaire)
 - Listes de médicaments par poste
 - Temps de picking poste
- **Variables de sortie** :
 - Temps de préparation des médicaments
 - Utilisation des ressources (humains/robots)
- **Hypothèses** :
 - Vitesse de déplacement dans le circuit constant
 - Il n'y a pas de rupture de stock
- **Fin de la simulation** à la distribution du dernier médicament.
- **Récupération / Tri des données** : Il n'y a pas de récupération de données puisque cette simulation a pour but d'être implantée dans une autre.

A partir de ce cadre expérimental, nous avons recherché les éléments du **MGC** que nous souhaitons utiliser. Cette pharmacie est composée d'une ligne d'assemblage que l'on peut décomposer en plusieurs entités du **MGC** :

- une station de travail équipée d'un robot,
- une station de travail avec un préparateur,
- un stock de médicament,
- un stock de *Doses Préparées pour un Patient* (**DPP**) rangées par service puis par pôle.

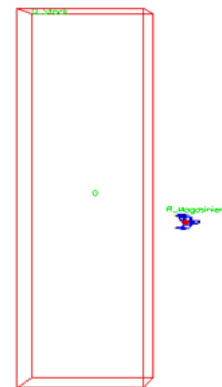
Pour étudier les relations et les interactions dans une pharmacie générique avec une dispensation nominative, nous avons utilisé le diagramme des **ED** de la figure 6.12.

Pour pouvoir respecter ce cadre expérimental, nous avons conçu un modèle qui comporte 5 étapes que nous avons récapitulées dans la table 6.3.

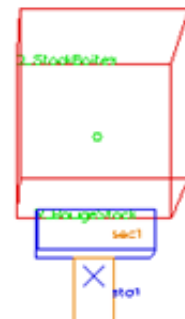
Nous avons conçu les éléments de ce modèle suivant l'analyse des différents systèmes d'assemblage (en série, avec ou sans dérivation) que nous avons effectué à la section 1.4. Au début, les médicaments se trouvent dans le stock de la pharmacie, un opérateur (préparateur ou magasinier) est là pour approvisionner les différents postes du circuit. Le stock de boîtes vides envoie alors des boîtes sur le convoyeur. Elles sont acheminées aux différents postes souhaités ; les médicaments sont alors ajoutés dans les boîtes. Quand tous les médicaments sont dans la boîte, celle-ci va au poste de sortie. La boîte est alors prise par un opérateur qui la range dans le stock correspondant à son service.

Lors de la phase d'écriture des caractéristiques d'un système, il devra être décidé du nombre de postes de travail équipés de robot et du nombre de postes avec une armoire gérée par les préparateurs. Il y aura obligatoirement un stock de médicaments à la pharmacie et un poste de sortie pour le rangement par patient. Un exemple d'utilisation de cette pharmacie générique est donné au chapitre suivant. Les parties de code les plus importantes de ce modèle sont expliquées dans l'annexe A.3.

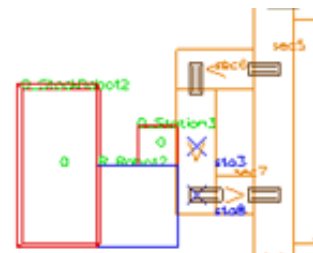
Stock de la pharmacie



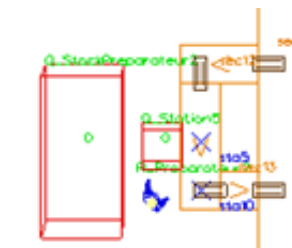
Approvisionnement du circuit avec les boîtes vides



Poste de travail avec un robot



Poste de travail avec une armoire de stockage gérée par un préparateur



Poste de sortie

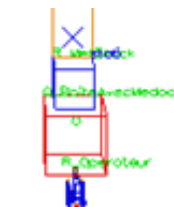


TABLE 6.3 – Récapitulatif de la représentation des services, des pôles et des patients sous Automod, en DJIN et en dispensation globale

6.8.4.3 Représentation des stocks de médicaments

Dans le cas de la **DJIN**, nous allons livrer un médicament à l'intérieur d'un tiroir pour chaque patient. Chaque tiroir sera à l'intérieur de l'armoire d'un service, qui sera lui-même à l'intérieur d'un pôle. La représentation par *container* s'effectue alors ainsi : le médicament est placé dans un *sous-container* (modélisant un tiroir pour chaque patient et représenté par un cube bleu). Nous avons représenté dans la table 6.4 la représentation que nous avons faite pour une dispensation nominative et globale/globalisée.

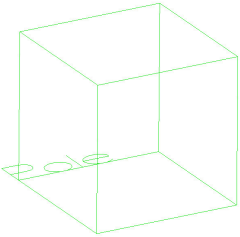
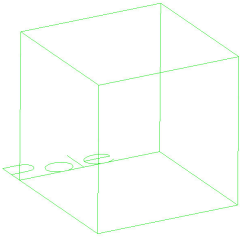
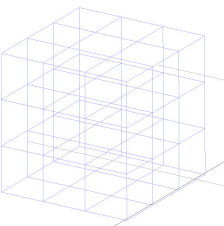
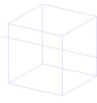
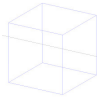
	DJIN	Dispensation globale et globalisée
Pôle	Queue 	Queue 
Service	Container 	Sous-Container 
Patient	Sous-Container 	Non représenté

TABLE 6.4 – Récapitulatif de la représentation des services, des pôles et des patients sous Automod, en DJIN et en dispensation globale/globalisée

Le *sous-container* est placé dans un *container* (modélisant l'armoire d'un service). Les *containers* sont placés dans une *Queue* (modélisant un pôle et représentée par un cube vert). Dans le cas d'une dispensation globale ou globalisée, la représentation est plus simple, car nous n'avons pas besoin de représenter les médicaments pour chaque patient. De ce fait, nous n'avons plus qu'un *container* par *queue*. Le *container* (en bleu) représente le pôle correspondant et chaque *sous-container* un service.

6.8.4.4 Interface d'entrée-sortie

Autostat est un module d'AutoMod qui permet de traiter des données et de faire des analyses statistiques sur un modèle créé avec AutoMod. En analysant un modèle, Autostat permet de déterminer une période de warm-up ou de calculer des intervalles de confiance. Cette interface rend possible la conception d'un véritable outil d'aide à la décision puisqu'il permet de comparer rapidement plusieurs scénarios ou plusieurs configurations.

L'essentiel d'Autostat se résume avec quatre onglets :

- Factors : on définit les facteurs (les variables d'entrée) à faire varier,

- Responses : permet de définir les variables de sortie,
- Runs : permet d'effectuer les itérations, en définissant les conditions de celles-ci,
- Analyses : permet de créer une configuration de scénario, en sélectionnant les variables d'entrée et de sortie préalablement définies ainsi que leur valeur et d'obtenir les résultats.

Une autre interface d'entrée/sortie a été développée sous Excel ; elle permet de gérer les fichiers en entrée de différents logiciels de simulation (Automod et Witness) dont on peut voir une capture d'écran à la section 6.6.

6.9 PHASE 2 - BILAN

Notre approche nous fournit un cadre méthodologique qui nous a permis de dérouler les étapes de conception et d'implémentation de manière structurée et cohérente.

En effet, à partir du **MGC** construit dans la phase 1 et des étapes de Conception et d'Implémentation de la phase 2, nous avons développé une bibliothèque de composants génériques. Ces étapes ont permis de décrire précisément le fonctionnement des holons et de leurs interfaces que nous avons préalablement décrits dans le **MGC**. Nous avons montré dans les sections précédentes trois études.

La première étude a montré une approche de modélisation pour la conception d'une interface d'entrée/sortie pour des logiciels de simulation. Nous nous sommes focalisés sur le holon chariot. Nous avons représenté le comportement de cet holon lors de sa création dans l'interface d'entrée pour permettre son utilisation dans un logiciel de simulation. L'étape d'Implémentation a consisté en le codage de ce diagramme dans l'interface.

La deuxième a été l'étude de l'interface de validation entre le **SI** d'un service et les **DPP** pour réaliser la conception d'un logiciel permettant la validation de l'administration par l'infirmier au lit du malade, via le **SI** et l'utilisation de tags **RFID** dans une architecture contrôlée par le produit. Nous avons effectué cette étude avec des diagrammes des cas d'utilisation de **SysML**, comme le préconisent plusieurs méthodologies. Ensuite, dans l'étape d'Implémentation, nous avons codé le logiciel.

Enfin, la dernière étude a permis la conception de deux pharmacies génériques, qui sont modélisées dans un logiciel de simulation. La première porte sur la dispensation globale et globalisée : elle découle du diagramme des **ED** représentant le contrôle par les ressources. La deuxième représente une pharmacie avec une dispensation nominative : elle découle du diagramme des **ED** représentant un contrôle par le produit. Il est à noter que ces pharmacies génériques, du fait justement de leur genericité, sont compatibles, quel que soit l'hôpital étudié.

6.10 GÉNÉRICITÉ DE L'APPROCHE PROPOSÉE

Dans la phase 1, nous avons construit un *Modèle Générique de Connaissance* (**MGC**) du domaine du **CDM**. Ce modèle est générique car il est utilisable pour n'importe quel établissement de santé en France. Nous l'avons construit en partenariat avec des pharmaciens, et après avoir étudié le fonctionnement des autres hôpitaux français. Les composants faisant partie de la bibliothèque de composants réutilisables sont aussi génériques car ils ont été construits dans l'optique d'être adaptables pour différents **SI**. La seule condition est donc que les établissements soient équipés d'un **SI**.

Si nous reprenons le composant de suivi du **CDM** par **RFID**, cette application ne fonctionne pas avec un **SI** particulier mais peut s'adapter au **SI** de tous les hôpitaux. En effet, les données dont l'application a besoin sont disponibles dans tous les hôpitaux. Il faudra juste reprendre le modèle relationnel et y adapter les données.

Pour le cas d'étude de l'interface d'entrée/sortie, les composants constituant l'interface sont les mêmes dans tous les hôpitaux. Cette interface est donc adaptable directement.

Pour le cas d'étude des modèles de simulation d'une pharmacie générique, il faut insérer cette pharmacie dans le modèle d'un hôpital existant ou à construire. On peut alors évaluer les performances en mettant en place différents scénarios (nombre de ressources, temps de cycle...). On pourra alors comparer différents types de fonctionnement au niveau de la pharmacie et ainsi réorganiser au mieux l'hôpital.

6.11 CONCLUSION

Ce chapitre a présenté les deux phases qui permettent la conception d'un *Modèle Générique de Connaissance* (**MGC**) et une bibliothèque de composants réutilisables du *Circuit Du Médicament* (**CDM**). Ces deux phases font partie de la *Méthodologie de Réingénierie des HMS* (**MRHMS**) que nous avons définie au chapitre 5. Elles suivent la méthodologie *ASCI pour la Réingénierie des HMS* (**ASCI-RHMS**) que nous avons appliquée au **CDM**. Nous avons appelé la méthodologie *ASCI pour la Réingénierie des HMS* (**ASCI-RHMS**) appliquée au domaine du **CDM**, la méthodologie **ASCI-CDM**.

La première phase de cette méthodologie correspond aux étapes d'Analyse et de Spécification. Dans l'étape d'Analyse, nous avons effectué une décomposition systémique du **CDM** formalisée par des diagrammes de classe **UML** en reprenant et augmentant la proposition faite par [El Haouzi 08a, El Haouzi 08b] pour un système d'assemblage. Nous avons ensuite décrit le comportement général d'un **CDM** lors de l'étape de Spécification avec des diagrammes d'activité **UML**. Nous avons enrichi la décomposition systémique par deux *Internal Block Diagram - diagramme interne de bloc* (**IBD**) du langage **SysML** qui font le lien entre les sous-systèmes physique et décisionnel de cette décomposition. Ces diagrammes représentent les *Entités Décisionnelles* (**ED**) et permettent de représenter le système dans le cas d'une dispensation globale et globalisée (préparation par service) et dans le cas d'une dispensation nominative (préparation par patient). La dispensation nominative est le fonctionnement le plus sécurisé du **CDM** et nous souhaitons utiliser le concept de *Système Contrôlé par le Produit* (**SCP**) pour pouvoir l'implanter. L'utilisation de ce concept permet d'avoir une plus grande flexibilité et modularité. Le langage **SysML** est préconisé par la méthodologie **MBCSA** qui est constituante de **ASCI-RHMS**. Il permet de décomposer aisément chaque composant en autres composants à un niveau d'abstraction inférieur. Il permet aussi de représenter les interfaces entre composants. L'objectif est de pouvoir isoler dans la deuxième phase les composants que nous devons modifier et étudier pour permettre la réingénierie.

La deuxième phase est simplifiée par les deux diagrammes des **ED** qui permettent d'isoler les composants et les interfaces à modifier et à étudier pour un sujet d'étude donné. Nous avons pu décrire les éléments à analyser et spécifier pour la Conception et l'Implémentation d'une interface d'entrée/sortie pour des logiciels de simulation. Nous avons aussi étudié l'interface de vérification pour effectuer un dispositif de vérification au lit du malade en s'aidant des étapes de Conception de d'Implémentation d'**ASCI-CDM**. Et enfin, nous avons conçu deux pharmacies génériques qui correspondent aux deux types de livraison, c'est-à-dire une avec une dispensation globale et globalisée et une avec une dispensation nominative. Le but

est de permettre de les insérer dans un modèle de simulation d'un hôpital et d'évaluer les performances entre ces différents types de dispensation pour mieux réorganiser une pharmacie. Tous ces composants construits seront ajoutés dans une bibliothèque de composants réutilisables.

Nous avons donc effectué un **MGC** d'un domaine : le **CDM**. Le chapitre 7 présentera à partir de ce **MGC** une étude de scénario sur un système réel en suivant les recommandations de la méthodologie **MRHMS**. Les modèles de simulation présentés dans ce chapitre seront insérés dans le modèle de simulation d'un hôpital pour étudier différents types de fonctionnement et ainsi permettre la réingénierie de cet hôpital.

7

INSTANCIATION SUR UN SYSTÈME : LE CHU DE CLERMONT-FERRAND

RÉSUMÉ DU CHAPITRE 7

Le circuit du médicament peut être vu comme un système d'assemblage. Nous avons proposé un modèle générique de connaissance du domaine du circuit du médicament dans le chapitre précédent. La construction de ce modèle générique de connaissance a été permise grâce à l'utilisation de notre méthodologie MRHMS que nous avons proposée dans le chapitre 5. Elle a plusieurs finalités :

- elle peut être utilisée du « Méta-modèle pour la réingénierie » au « Modèle de connaissance d'un système » ; il s'agit alors d'une étude complète (nous entendons par étude complète l'étude du domaine + l'étude d'un système bien particulier de ce domaine) ;*
- l'analyste peut n'avoir qu'à étudier uniquement un domaine sans s'intéresser à un système particulier ; il pourra alors s'appuyer sur notre méthodologie (cette partie fait l'objet du chapitre 6) ;*
- si une étude préalable s'est déjà intéressée au domaine du système, notre méthodologie peut alors poursuivre cette étude vers l'étude du système particulier (ce qui est présenté dans ce chapitre).*

Dans ce chapitre, nous proposons de reprendre le modèle générique de connaissance relatif au domaine du circuit du médicament construit dans le chapitre précédent et de l'instancier sur un système avec la mise en place d'un scénario au CHU de Clermont-Ferrand. Pour ce faire, notre méthodologie ASCI-CDM, composante de MRHMS, préconise quatre étapes :

- 1. une étape de définition des caractéristiques du système avec la mise en place d'un scénario que nous voulons étudier à travers un langage de modélisation de processus,*
- 2. une étape de construction du modèle de connaissance. Nous instancions les diagrammes représentant les entités décisionnelles avec ce scénario d'étude,*
- 3. une étape d'élaboration du Modèle d'Action avec la mise en œuvre de ce scénario. Nous utilisons les modèles de pharmacies génériques développés sous un logiciel de simulation au chapitre 6,*
- 4. une étape d'élaboration du modèle de résultats pour permettre une évaluation des performances du modèle d'action. Nous utilisons alors une interface d'entrée/sortie pour présenter nos résultats.*

SOMMAIRE

7.1	INTRODUCTION	137
7.2	CARACTÉRISTIQUES D'UN SYSTÈME	139
7.2.1	Définition des caractéristiques du système	139
7.2.2	Le langage utilisé	139
7.2.3	À partir des diagrammes	141
7.2.4	À partir des matrices de déroulement	141
7.2.5	Étude de scénario avec la pharmacie nominative générique	143
7.3	MODÈLE DE CONNAISSANCE DU SYSTÈME	144
7.3.1	Le SIH au CHU de Clermont-Ferrand	144
7.3.2	Modèles pour la réingénierie à partir du MGC	146
7.3.2.1	Construction du Modèle de Connaissance	146
7.3.2.2	Fonctionnement actuel du CDM au CHU de Clermont-Ferrand	146
7.3.2.3	Fonctionnement souhaité du CDM au CHU de Clermont-Ferrand	146
7.4	MODÈLE D'ACTION	148
7.4.1	Élaboration du modèle d'action	148
7.4.2	Un nouvel hôpital	149
7.4.2.1	L'hôtel Dieu	149
7.4.2.2	Le CHU Estaing	149
7.4.3	Modélisation du CHU Estaing	151
7.4.3.1	La distribution	151
7.4.3.2	Rangement des UF	152
7.4.3.3	La pharmacie du CHUE	152
7.4.4	Interface d'entrée	153
7.4.4.1	Hypothèses de fonctionnement	153
7.4.4.2	Listes des doses à préparer et évaluation du volume des médicaments transportés	154
7.4.4.3	Attribution des UF par tournée	155
7.4.5	Ajout de la pharmacie générique pour une livraison nominative	155
7.4.6	Bilan et exploitation du modèle d'action	157
7.4.6.1	Correspondance avec le MC et perspectives du modèle d'action	157
7.4.6.2	Cadre expérimental	158
7.5	MODÈLE DE RÉSULTATS	158
7.5.1	Mise en œuvre du modèle d'action	158
7.5.2	Alimentation du modèle de résultats	160
7.6	CONCLUSION	162

7.1 INTRODUCTION

Le chapitre 6 a montré une application de notre méthodologie **MRHMS** présentée au chapitre 5 sur le domaine du *Circuit Du Médicament* (**CDM**). Elle nous a permis, grâce à l'utilisation d'*ASCI pour la Réingénierie des HMS* (**ASCI-RHMS**) instancié au *Circuit Du Médicament* (**CDM**) (**ASCI-CDM**), la construction d'un *Modèle Générique de Connaissance* (**MGC**). Dans ce chapitre, nous instancierons les modèles et composants élaborés pour ce domaine à un système particulier de celui-ci : le **CHU** de Clermont-Ferrand. Le positionnement du chapitre 7 par rapport à la méthodologie **MRHMS** est montré sur la figure 7.1.

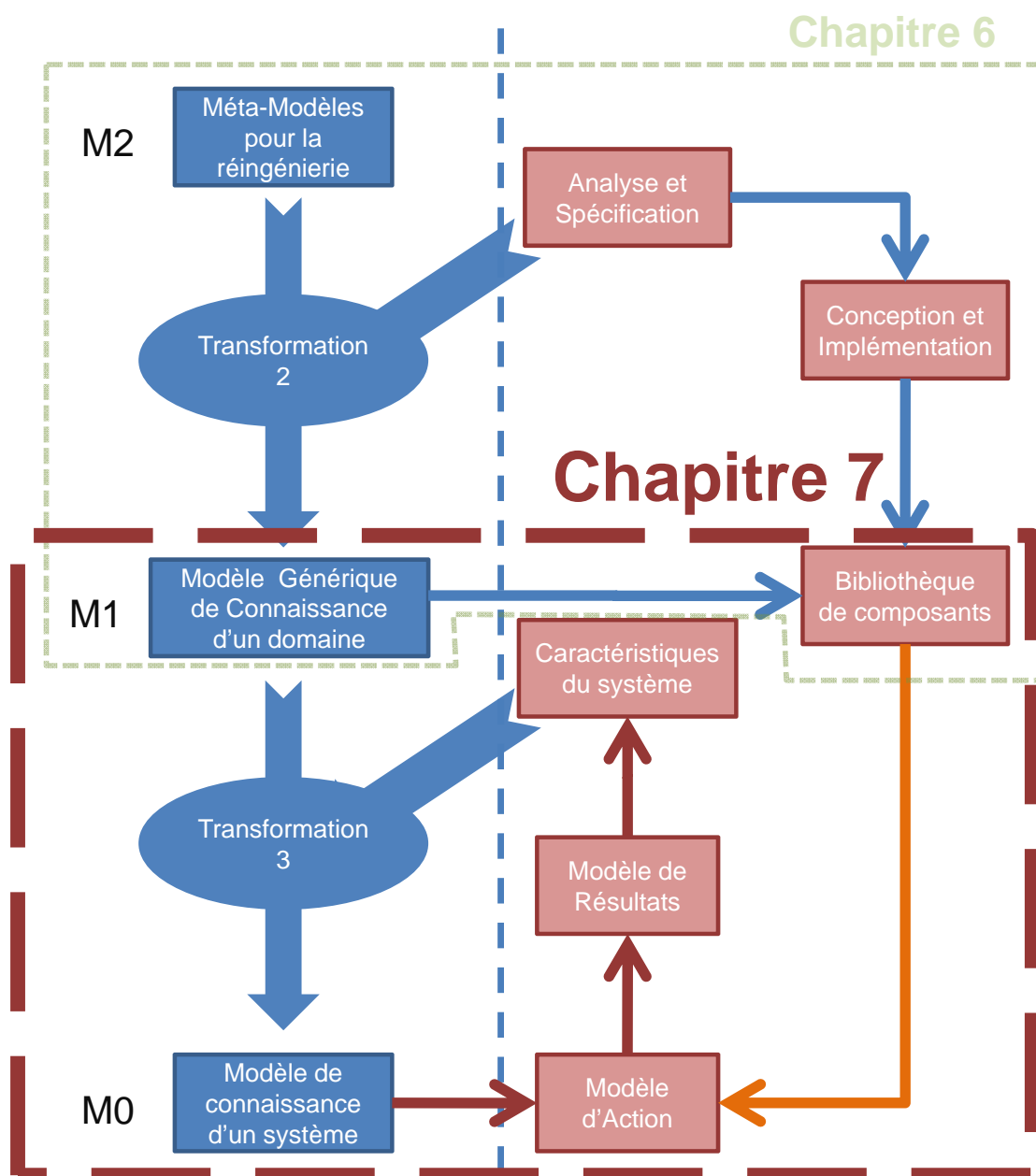


FIGURE 7.1 – Positionnement du chapitre 7 dans le mémoire

Le **MGC** construit dans le chapitre précédent nous a permis d'étudier tous les types de préparation, de livraison et de rangement sur le **CDM**. A partir de ce **MGC**, nous allons élaborer le modèle de connaissance d'un système de ce domaine. Cette élaboration consistera en l'instanciation des holons du diagramme des *Entités Décisionnelles* (**ED**) que nous avons établis dans le chapitre précédent. Ces diagrammes montrent le **CDM** lors d'une dispensation globale et lors d'une dispensation nominative. La dispensation globale est conforme au méta-modèle représentant un contrôle par les ressources et la dispensation nominative est conforme au méta-modèle représentant un contrôle par le produit.

Le processus de modélisation préconisé par la méthodologie **ASCI-RHMS** pour l'étude d'un système du domaine est composé de quatre étapes :

1. une étape décrivant les caractéristiques du système à étudier (section 2). Nous représenterons les éléments constituant le système à étudier. Nous utiliserons un langage qui modélise aisément un processus industriel,
2. l'étape de construction du *Modèle de Connaissance* (**MC**) en instanciant les holons du **MGC** (section 3),
3. l'étape d'élaboration du *Modèle d'Action*. Le modèle d'action est une traduction partielle ou complète du **MC** dans un langage de programmation (langage général de programmation, langage général de simulation ou simulateur dédié) permettant l'évaluation des critères de performance choisis qui constituent le modèle de résultats. Dans la section 4, nous montrerons le *Modèle d'Action* généré sous le logiciel de simulation Automod,
4. l'étape d'évaluation des performances dans le *Modèle de Résultats* qui sera alimenté par le *Modèle d'Action*. Nous utiliserons, dans la section 5, le module Autostat de Automod pour exploiter le modèle de simulation.

7.2 CARACTÉRISTIQUES D'UN SYSTÈME

7.2.1 Définition des caractéristiques du système

Selon **ASCI-RHMS**, cette étape doit être élaborée en collaboration avec des experts du système étudié. Pour **ASCI-CDM**, le système doit alors être construit en collaboration avec les pharmaciens et la direction du **CHU**. Les caractéristiques du système doivent permettre la mise en place de scénarios et pouvoir les détailler à travers une représentation claire. Nous proposons alors d'utiliser le langage décrit par la norme JIS Z 8206. Ce formalisme a l'avantage d'être utilisé soit avec un diagramme, soit avec une représentation sous forme de matrices qui permet de détailler chaque élément du diagramme.

7.2.2 Le langage utilisé

Pour la transformation entre le niveau M_1 à M_0 , nous effectuons une instanciation telle que définie dans les chapitres précédents dans la méthodologie **ASCI-RHMS**. Pour réussir correctement cette transformation, nous devons identifier les composants qui seront présents dans le système ainsi que les relations entre eux. Nous avons alors cherché un outil qui permet de modéliser facilement le comportement d'un système.

Pour la modélisation d'un processus d'un **CDM**, nous avons utilisé le formalisme de la norme JIS Z 8206 décrit dans la section 3.5.4 qui permet d'obtenir facilement un modèle synthétique et général des flux physiques. La table 7.1 montre la signification des symboles donnés dans [Lee 99] et leurs utilisations pour le **CDM**. Des matrices et diagrammes basés sur des symboles standardisés décrivent le flux du processus et permettent de déterminer rapidement où les problèmes peuvent se poser dans le processus. Les symboles sont divisés en symboles basiques et supplémentaires. Les graphiques basiques consistent en six symboles qui sont l'opération d'addition, le transport, le stockage, la temporisation, l'inspection du volume et l'inspection de la qualité. Nous les avons montrés dans la table 7.1. Nous comparons le produit d'un système de production à l'ensemble des *Doses Préparées pour un Patient* (**DPP**) sur le **CDM**. Nous avons présenté l'utilisation de cet outil dans [Huet 09b, Huet 09c].

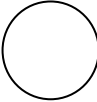
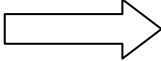


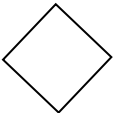

Etape	Symbole	Signification [Lee 99]	Signification pour notre système
Opération		Modifie la forme ou d'autres caractéristiques d'une pièce ou d'un produit	Opération d'ajout de médicaments dans des DPP
Transport		Change l'emplacement des pièces	Déplacement des DPP
Stockage		Accumulation programmée de pièces ou de produits	Accumulation programmée des ensembles de DPP
Temporisation		Accumulation non-programmée de pièces ou de produits	Accumulation non-programmée de préparation de prescription
Inspection de la qualité		Test et inspection visuel de pièces ou produits en comparaison avec la qualité standard pour juger si un produit défectueux (non-standard) a été produit	Test et inspection visuel de la préparation avec la prescription informatisée pour juger si une préparation défectueuse a été produite
Inspection du volume		Mesure de la quantité de pièces et de produits pour les comparer avec la quantité spécifique pour juger si une divergence existe	Mesure de la quantité de médicaments et de DPP avec la prescription et prise de décision sur la façon de continuer le remplissage

TABLE 7.1 – Signification des symboles de la norme JIS Z 8206 et leur correspondance sur le circuit du médicament

Chaque étape de la préparation est détaillée dans la matrice de déroulement dont on trouvera un exemple en section 7.2.4. De plus, cette matrice peut donner le flux avec les unités, une section, un département ou entre départements. Ce flux peut inclure des étapes séquentielles d'opérations de production, où il peut inclure les étapes séquentielles que des opérateurs humains peuvent avoir à réaliser. Le diagramme de déroulement complète la matrice de déroulement. Il est utilisé pour étudier chaque étape en dessinant la mise en place de chaque zone dans laquelle le processus se déroule. Cette mise en place sera ainsi compréhensible plus facilement en utilisant les mêmes symboles utilisés que dans la matrice de déroulement. Ceci est utilisé par exemple dans [Lee 99], où on peut voir une figure qui illustre un diagramme de déroulement qui décrit le chemin, la séquence et les distances pour un processus.

7.2.3 À partir des diagrammes

Nous avons recherché à représenter les séquences qui correspondraient à un CDM à partir de cette analogie et à partir de l'état de l'art sur les différents types de systèmes d'assemblage donné à la section 1.4. Après discussions avec les pharmaciens, nous avons considéré que l'architecture suivante est la plus représentative d'une dispensation nominative.

Nous donnons sur la figure 7.2 un exemple d'application qui représente les trois séquences de base pour modéliser un système d'assemblage tel que nous l'avons défini.

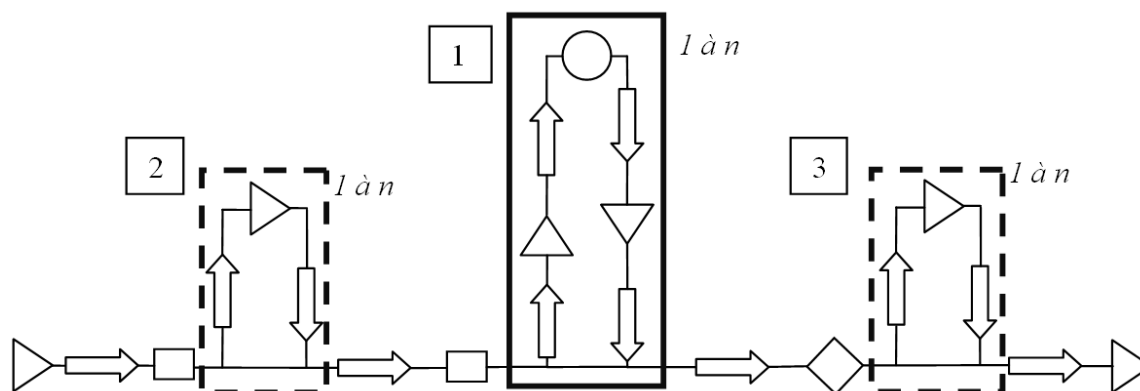


FIGURE 7.2 – Modélisation possible du circuit du médicament

Cette figure est décomposée en trois séquences :

- Le rectangle avec les lignes pleines (rectangle 1) est un modèle d'opération d'assemblage,
- Les rectangles avec les lignes pointillées (rectangles 2 et 3) sont les modèles d'un stockage,
- Le reste : des éléments simples de notre processus.

Nous avons vu qu'une opération d'assemblage peut être comparée à un ajout de médicaments. Il peut être effectué par un préparateur ou par un robot. Chaque séquence peut être répétée de 1 à n fois. n dépend de l'organisation choisie par les pharmaciens suivant les ressources dont ils disposent. Comme montré sur le rectangle 1 (figure 7.2), une opération d'ajout d'un ensemble de médicaments, qu'elle soit exécutée par un opérateur humain ou par un robot, est composée de trois étapes :

- Un stockage,
- Une opération d'assemblage,
- Un stockage.

Et entre ces étapes, nous avons un transport dans les différentes zones.

Les explications détaillées sont données avec les matrices de déroulement dans la partie suivante.

7.2.4 À partir des matrices de déroulement

Dans le but d'avoir plus de détails, nous avons construit une matrice comme [Fontanili 99] et [Lee 99] l'ont définie. Les tables 7.2, 7.3 et 7.4 détaillent ce que le système fait à chaque étape du diagramme de la figure 7.2. Le « conteneur » est l'entité où nous stockons l'ensemble des boîtes qui contiennent les DPP. La table 7.2 donne la description de chaque étape du chemin principal. Les détails de l'opération dans le rectangle 1 et dans les rectangles 2 et 3

sont décrits respectivement dans les tables 7.3 et 7.4. La colonne « étape » explique chaque symbole contenu dans la figure 7.2.





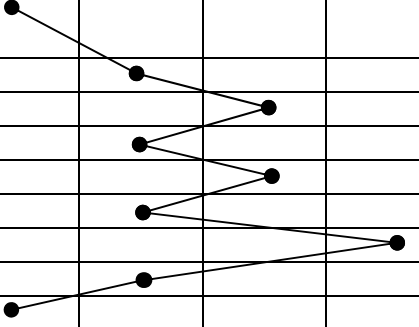
Symbole				Etape
				
				Stockage du conteneur, commencement d'une nouvelle production
				Le produit est dans le conteneur et sur le convoyeur
				Décision du lieu où doit aller le produit
				Dans le conteneur, vers le convoyeur
				Décision du lieu où doit aller le produit
				Sur le convoyeur
				Test de la qualité et comparaison avec la prescription
				Transport avec pneumatique ou chariot
				Stockage final, dans les unités de soin

TABLE 7.2 – Matrice de déroulement du chemin principal sans aucune opération d'assemblage



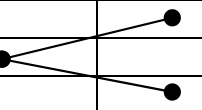
Symbole		Etape
		
		Du convoyeur vers la zone de stockage
		Dans la zone de stockage
		Sur le convoyeur

TABLE 7.3 – Matrice de déroulement pour atteindre une zone de stockage




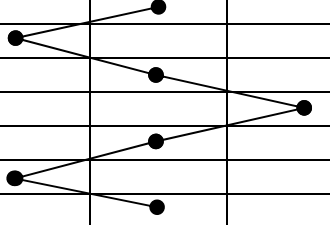
Symbole			Etape
			
			Du convoyeur vers la zone de stockage
			Dans la zone de stockage de la ressource
			De la zone de stockage vers la ressource
			Ajout d'un ensemble de médicaments (robot ou préparateur)
			A la fin de l'opération ; vers la zone de stockage
			Dans la zone de stockage de la ressource
			De la zone de stockage de l'opération vers le convoyeur

TABLE 7.4 – Matrice de déroulement pour une opération d'assemblage

Un scénario complet est composé d'une séquence représentant le chemin principal (table 7.2) et de plusieurs fois les deux autres séquences (1 à n fois). Les tables 7.3 et 7.4 représentent les chemins qu'il faut ajouter au chemin principal et correspondent respectivement aux rectangles 2 et 1 de la figure 7.2. Le chemin complet dépend de ce que les pharmaciens souhaitent étudier comme équipement pour leurs pharmacies.

Par conséquent, le chemin final est composé par le chemin principal et 1 à n fois les autres chemins représentés par les tables 7.3 et 7.4 et les rectangles 1, 2 et 3 de la figure 7.2.

7.2.5 Étude de scénario avec la pharmacie nominative générique

Dans le chapitre précédent, nous avons réalisé deux modèles de simulation de pharmacies génériques. Un avec une dispensation nominative utilisant le concept de système contrôlé par le produit et une avec une dispensation globale utilisant le concept de système contrôlé par les ressources. Une pharmacie avec une dispensation nominative est composée de plusieurs éléments décrits dans la section 6.8.4.2 : un stock de médicaments, une station d'approvisionnement du circuit avec les boîtes vides, de 1 à n postes de travail avec un robot, de 1 à n postes de travail avec une armoire de stockage gérée par un préparateur et un poste de sortie. Les postes de travail correspondent aux opérations d'assemblage (rectangle 2) et le poste de sortie à un stockage (rectangle 2 et 3). La station d'approvisionnement du circuit en boîtes vides correspond au premier symbole de stockage du chemin principal et le stock de médicament n'est pas représenté car nous ne montrons ici que le flux physique des DPP et non l'approvisionnement des stations en médicaments.

Nous avons étudié une pharmacie avec deux postes de travail équipés de robots et deux postes avec des entrées manuelles. Le premier triangle représente le stock de boîtes vides. Après être passés par les différents postes de travail, les DPP sont transportées au point de vérification : elle peut être faite automatiquement par une technologie d'identification ou/et par un pharmacien suivant le degré d'équipement de l'hôpital. Après cette dernière vérification à la pharmacie, les DPP sont stockées en attendant d'être transportées automatiquement ou par chariot. Nous le présentons sur la figure 7.3.

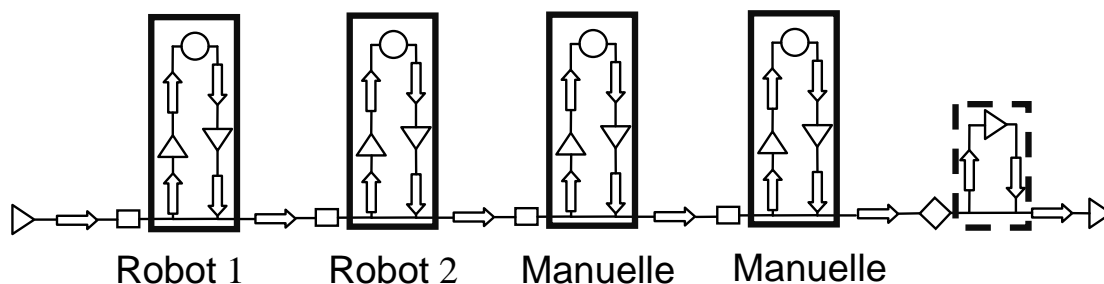


FIGURE 7.3 – Scénario pour l'étude de la pharmacie générique

A partir de ce diagramme, nous avons alors pris les éléments de cette pharmacie générique et nous les avons assemblés pour préparer la simulation. Nous obtenons alors la pharmacie nominative de la figure 7.4. Cette pharmacie générique est alors prête à être insérée dans le modèle d'un hôpital.

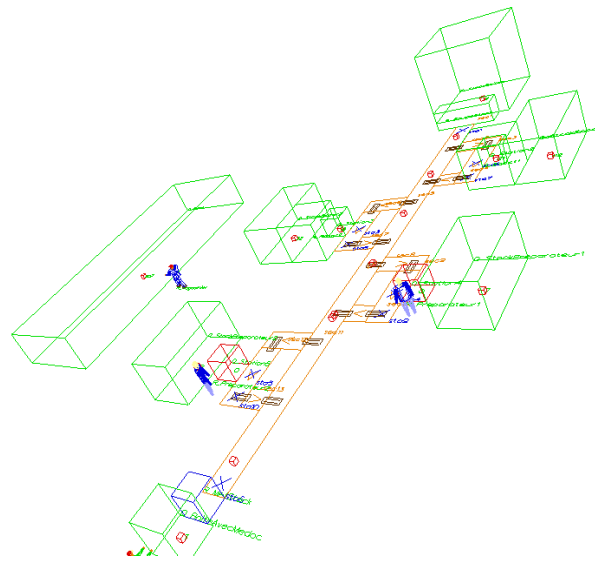


FIGURE 7.4 – Capture d'écran de la pharmacie générique avec le scénario étudié

7.3 MODÈLE DE CONNAISSANCE DU SYSTÈME

7.3.1 Le SIH au CHU de Clermont-Ferrand

Le **SIH** de l'hôpital de Clermont-Ferrand est composé de différentes sortes de **SI**. Les logiciels du **SIH** peuvent être divisés en différents modules qui sont utilisés avec différents **SI** de l'hôpital. Ici, nous nous intéressons au *Système d'Information Pharmaceutique* (**SIP**). Chaque logiciel utilisé peut aussi être divisé en modules qui comportent des applications spécifiques. Les **DPP** devront avoir des interactions avec différents logiciels suivant le moment de leur cycle de vie. Juste avant l'administration, le produit devra interagir avec le **SI** de l'unité de soins qui est utilisé pour gérer la prescription dans le but d'avoir une meilleure fiabilité.

Au CHU de Clermont-Ferrand, la majorité des services fonctionne avec une dispensation globale mais plusieurs services testent différents autres types de fonctionnement :

- Un service de cardiologie teste la dispensation globalisée avec prescription informatisée, analyse pharmaceutique de chaque prescription et validation informatique de chaque administration au patient par l'infirmier. C'est une première étape avant le passage à la dispensation nominative.
- Un service de chirurgie digestive teste une dispensation globale en système plein/vide avec déclenchement de la commande dans le **SIP** par puces **RFID**. Ce type de système restera pour les produits qui ne pourront pas être gérés en dispensation nominative.

Le logiciel utilisé pour la prescription informatisée est le logiciel CROSSWAY de la société Mc Kesson¹. Une capture d'écran de ce logiciel est montrée sur la figure 7.5. Elle représente un exemple de planning d'administration de médicament suite à l'écriture d'une prescription.

L'analyse pharmaceutique et la commande des services correspondent à deux modules d'un même logiciel : PHARMA de la société Computer Engineering². Une capture d'écran est donnée sur la figure 7.6. Elle représente le module d'analyse pharmaceutique. Ce logi-

1. <https://www.mckesson.fr/> (accessible au 8 octobre 2010)

2. <http://www.computer-engineering.fr> (accessible au 8 octobre 2010)

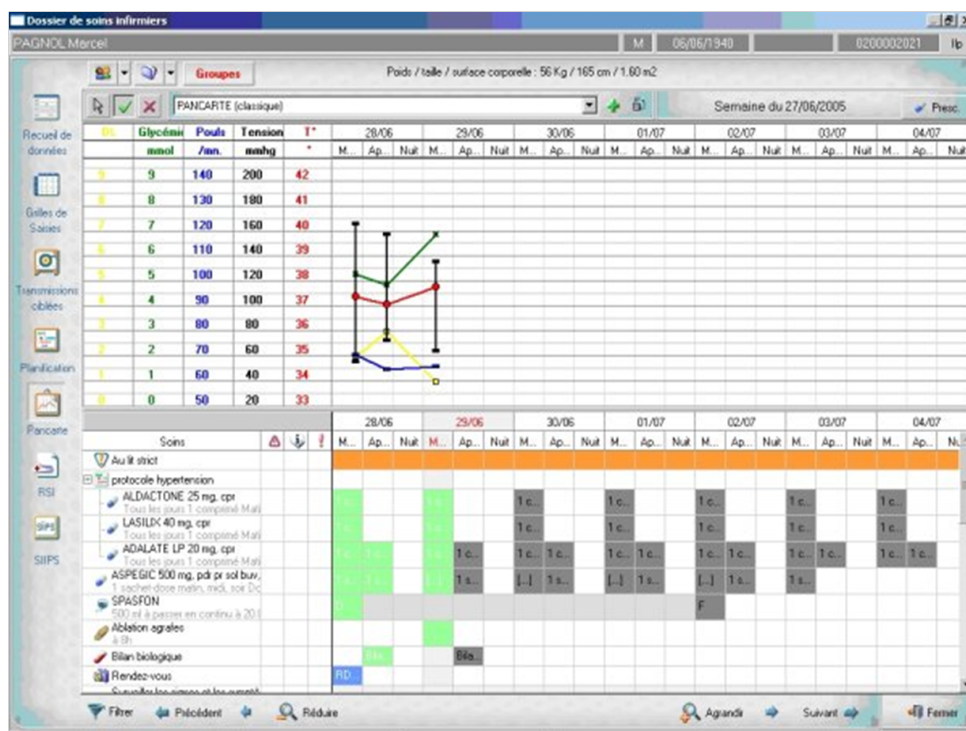


FIGURE 7.5 – Capture d'écran du logiciel de prescription informatisée

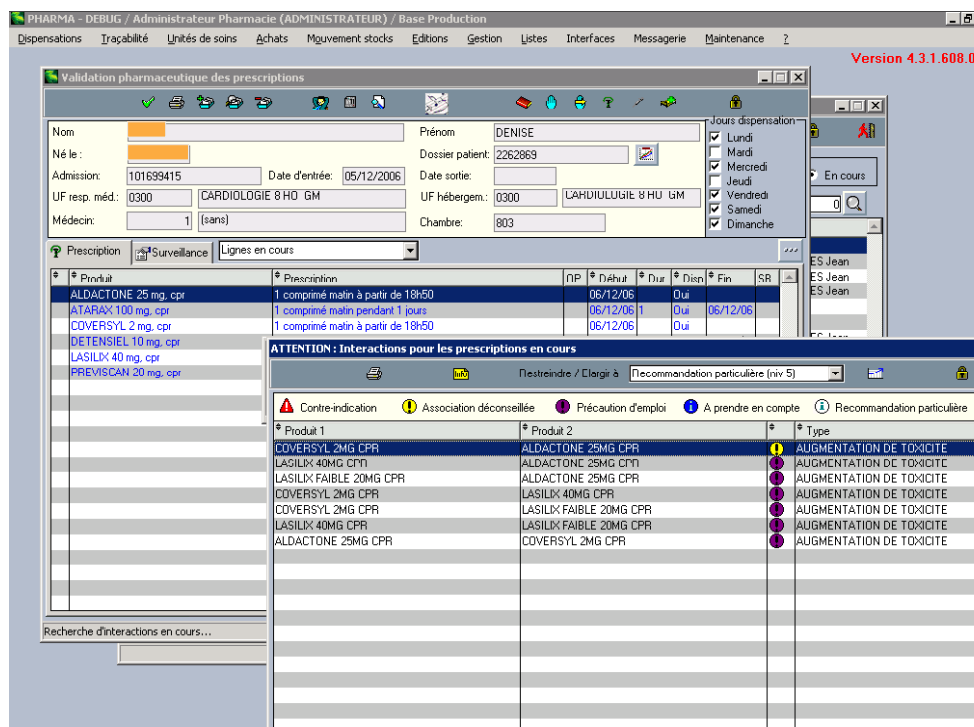


FIGURE 7.6 – Capture d'écran du logiciel de gestion des prescriptions au niveau de la pharmacie

ciel est aussi utilisé à la pharmacie pour gérer le stock de médicaments. Les données sont automatiquement transmises de CROSSWAY à PHARMA pour permettre la validation pharmaceutique. Le CHU de Clermont-Ferrand utilise aussi un circuit pneumatique pour envoyer des doses dans les unités de soins. Le logiciel utilisé par le circuit pneumatique est le logiciel WIN3000 de la société AEROCOM³. Ce logiciel permet de gérer le parcours d'une cartouche lors de son transfert dans le circuit. Le détail du circuit pneumatique utilisé au CHU de Clermont-Ferrand et sa représentation sous Automod sont donnés en annexe A.1.

7.3.2 Modèles pour la réingénierie à partir du MGC

7.3.2.1 Construction du Modèle de Connaissance

A partir de cette analyse du SIH du CHU de Clermont-Ferrand et des différents holons composant le CDM, nous pouvons maintenant instancier les diagrammes représentant les ED du MGC. Un de ces diagrammes représente une dispensation globalisée, il vient de l'instanciation du méta-modèle représentant un contrôle par les ressources d'un *Holonic Manufacturing System* (HMS). Le deuxième diagramme représente une dispensation nominative, il vient de l'instanciation du méta-modèle représentant un contrôle par le produit d'un *Holonic Manufacturing System* (HMS). Notre objectif est de pouvoir réorganiser un CDM en passant de l'un à l'autre. Nous allons alors instancier ces deux diagrammes à un cas d'étude. Nous nous trouverons alors dans le niveau Mo de notre méthodologie MRHMS, c'est-à-dire dans la modélisation d'un système du domaine.

7.3.2.2 Fonctionnement actuel du CDM au CHU de Clermont-Ferrand

A partir du fonctionnement actuel du CHU de Clermont-Ferrand, nous avons récupéré les éléments du MGC. Un module de PHARMA valide la fin de la préparation d'une commande pour une livraison dans un service, une feuille est mise dans le chariot à la destination de chaque service. Pour le circuit pneumatique, aucune feuille n'est imprimée, la préparation est juste validée et la cartouche transmise dans la station d'envoi. L'IBD de la figure 7.7 représente le fonctionnement pour une dispensation globalisée. Cela correspond au fonctionnement actuel le plus avancé de la pharmacie au CHU de Clermont-Ferrand au service cardiologie. Il est une instanciation du IBD de la figure 6.11 du chapitre 6 présentant une dispensation globalisée pour n'importe quel hôpital.

C'est donc notre point de départ pour la réingénierie du CDM d'un hôpital particulier, c'est à partir de ces éléments que nous devons atteindre la dispensation nominative en effectuant une transformation vers le modèle souhaité en identifiant les modifications à apporter sur les composants et leurs interfaces.

7.3.2.3 Fonctionnement souhaité du CDM au CHU de Clermont-Ferrand

Pour la mise en place d'un scénario d'étude, nous nous sommes alors inspirés d'autres hôpitaux français pour les éléments dont nous ne disposons pas. Nous avons visité le centre hospitalier d'Annecy et les hospices civils de Lyon qui utilisent les robots de dispensation nominative. Le choix de l'hôpital d'Annecy s'est porté sur les robots de la marque SYNTECO [Majchrzak 06]. Le SI de ce robot se nomme ATHENA. Le logiciel qu'il utilise dans la pharmacie pour gérer les stocks et les validations est le même qu'au CHU de Clermont-Ferrand : le logiciel PHARMA.

3. <http://www.aerocom-france.com> (accessible au 8 octobre 2010)

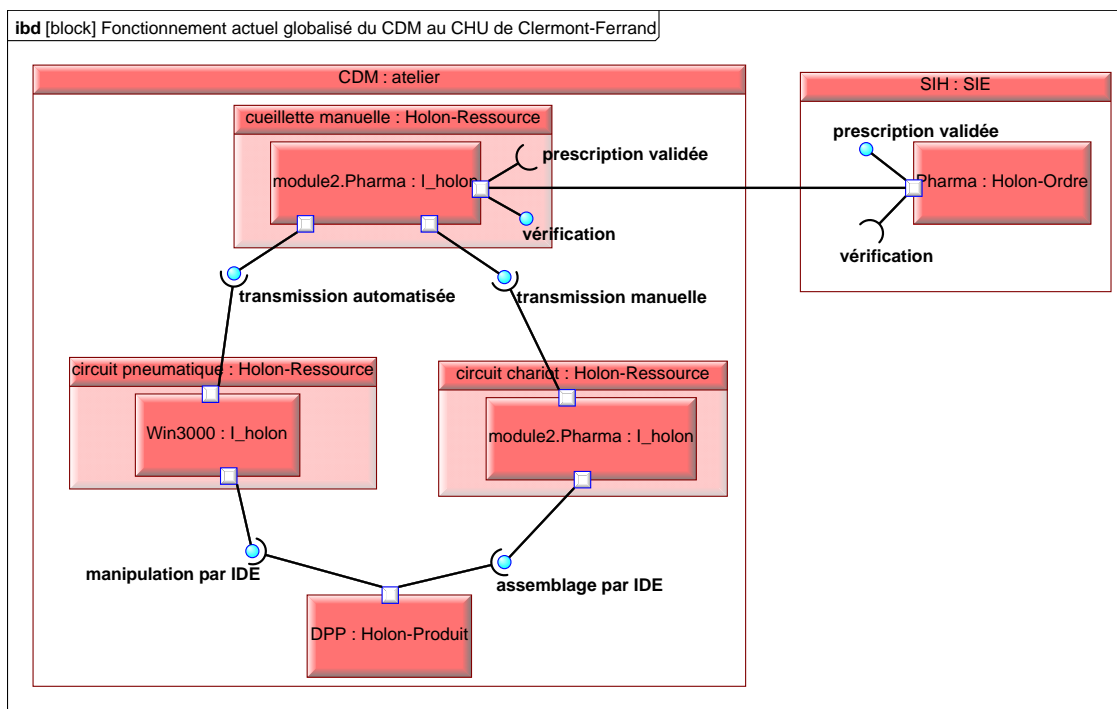


FIGURE 7.7 – Modèle de la dispensation globalisée au CHU de Clermont-Ferrand

Nous décidons d’instancier les holons du **MGC** représentant une dispensation nominative avec les composants présents au **CHU** de Clermont-Ferrand et le robot utilisé au CH d’Annecy. Le robot est instancié avec « SintecoRobot ». Cela signifie qu’il prend les attributs du robot de la marque Sinteco tel que le temps de cycle ou la liste des médicaments supportés. Le circuit automatisé correspond au circuit pneumatique. La « cueillette manuelle » ne change pas car il y a toujours une cueillette manuelle pour compléter le remplissage effectué par les robots. Le « Circuit Chariot » ne change pas car il y a toujours un circuit chariot classique. Le I_Holon (la partie informationnelle) du circuit chariot est aussi un module de PHARMA. A noter que le **HR** « circuit pneumatique » peut être décomposé lui-même en holons car les cartouches peuvent être utilisées avec des puces **RFID** pour vérifier leur parcours dans le circuit ; les cartouches pourront alors être étudiées dans une nouvelle phase de Conception et d’Implémentation pour alimenter la bibliothèque de composants.

Le diagramme est centré sur les interactions entre le **SIH** et le produit dans le pharmacie. Le fonctionnement des logiciels qui appartiennent à des sociétés, logiciels appelés « composants sur étagères » (Commercial off-the-Shelf - COTS) sont comme des boîtes noires indécomposables (on ne connaît pas leur fonctionnement interne). Par conséquent, nous nous concentrons uniquement sur la relation avec le produit. Le logiciel PHARMA est composé de différents modules utilisés avec différents composants. Le **HO** (un module de PHARMA ici) transmet la prescription validée à l’**HP** qui communique avec les **HR**. Les robots et les armoires communiquent aux **DPP** (le **HP**) les médicaments qu’ils peuvent ajouter. Le **HP** envoie des informations au circuit pneumatique et chariot à propos de ces choix d’outils de livraison pour le transport jusque dans l’unité de soins. En effet, le maximum de livraisons devront être effectuées par le circuit pneumatique qui a une meilleure flexibilité (horaires flexibles, automatisation de l’envoi...), cependant, certaines livraisons particulières (volume ou poids trop importants pour un circuit automatisé, frigorifiés, fragiles, etc..) devront toujours être

transportées par chariot. Le logiciel utilisé par le circuit pneumatique est le logiciel WIN3000, et le logiciel qui gère les armoires de stockage et le circuit chariot est un module de PHARMA. Pour mettre en place ce diagramme, nous instancions la la figure 6.12 avec tous ces éléments. Il est représenté sur la figure 7.8. C'est une vue « interopérabilité centré sur le produit », c'est-à-dire que nous ne montrons que le lien entre les parties informationnelles de chaque holon et destiné au contrôle par le produit.

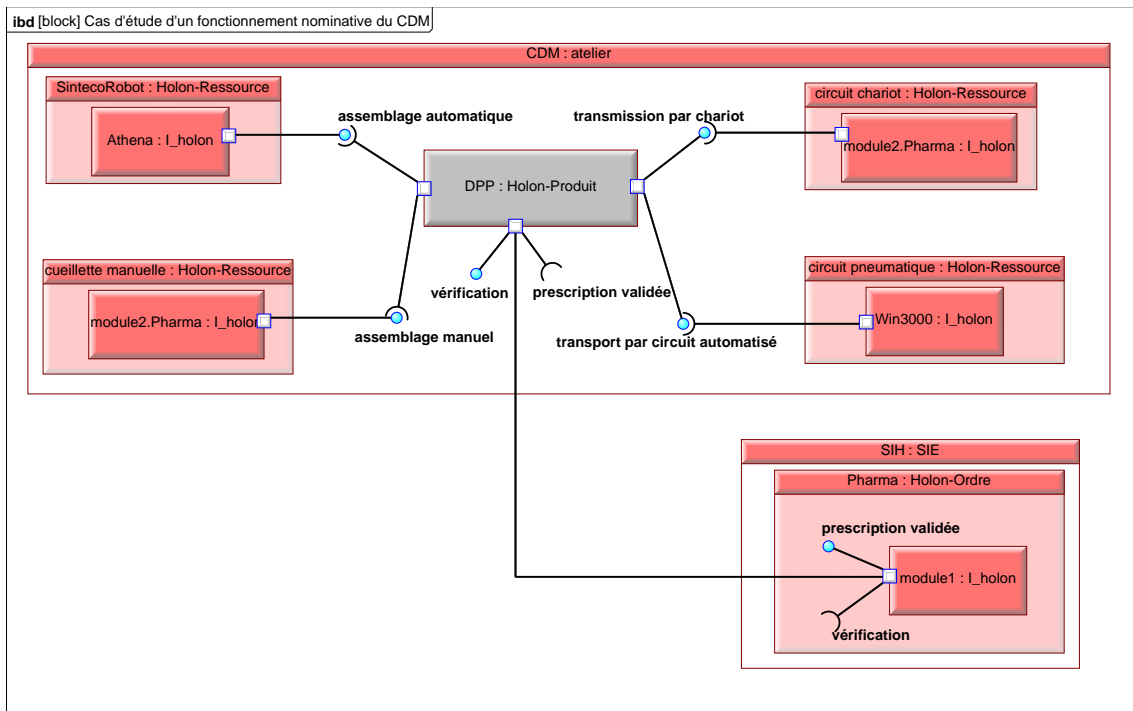


FIGURE 7.8 – Modèle d'une dispensation nominative au CHU de Clermont-Ferrand

7.4 MODÈLE D'ACTION

7.4.1 Élaboration du modèle d'action

De manière générale dans **ASCI**, un modèle d'action est défini comme une traduction partielle ou complète du modèle de connaissance (en fonction des objectifs et des hypothèses simplificatrices de la méthode utilisée) dans un formalisme mathématique avec des modèles génératifs (par exemple la programmation mathématique,...) et des modèles évaluatifs (par exemple les réseaux de file d'attente, les réseaux de Petri,...), ou dans un langage de programmation, langage général de simulation ou simulateur dédié.

Dans **ASCI-CDM**, le modèle d'action consistera en une traduction partielle du modèle de connaissance dans un langage de simulation. En effet, à partir du **MC** défini précédemment, nous pouvons représenter un hôpital équipé d'une pharmacie permettant la dispensation nominative et d'une pharmacie permettant la dispensation globale et globalisée. Nous décidons d'appliquer cette élaboration au modèle du CHU Estaing que nous présenterons dans les prochaines sections.

7.4.2 Un nouvel hôpital

7.4.2.1 L'hôtel Dieu

L'Hôtel-Dieu était le plus vieil hôpital du **CHU** de Clermont-Ferrand. Il a ouvert en 1773 sur l'emplacement de l'ancien Hôpital de la Charité. Il fut en son temps considéré comme exemplaire : œuvre d'architecture monumentale, il prévoyait en effet sur ses deux hectares d'espaces verts des lieux de promenade bien exposés pour faciliter la convalescence des malades. Ce n'est qu'à la fin du XIX^{ème} siècle qu'il devient un véritable hôpital médicochirurgical. Dans le cadre des grands travaux de restructuration du **CHU**, il avait été envisagé dans un premier temps de conserver sur son site quelques activités, et de créer une entité supplémentaire. Mais sur le plan de la rationalisation et de la mutualisation des moyens, cette hypothèse a été abandonnée au profit d'une autre : transférer l'ensemble des activités de l'actuel Hôtel-Dieu sur un nouveau site mieux adapté.

7.4.2.2 Le CHU Estaing

- La Médecine, la Chirurgie et l'Obstétrique (MCO) du **CHU** étaient installées sur 2 sites :
- L'Hôtel-Dieu en centre ville, construit en 1773, étendu à la fin du XIX^e puis dans les années 1920 / 1930, avec 530 lits et places.
 - L'Hôpital Gabriel Montpied avec 792 lits et places.

Si le site Gabriel Montpied conserve des possibilités d'évolution et d'extension, le site historique de l'Hôtel Dieu, principalement voué à la mère et à l'enfant, entièrement enclavé et en fort dénivelé, n'en possède pas. Par ailleurs, créer un troisième site MCO en conservant certaines activités à l'Hôtel-Dieu engendrerait de redoutables difficultés sur le plan de la rationalisation et de la mutualisation des moyens.

Le choix s'est donc imposé de transférer l'ensemble des activités de l'actuel Hôtel-Dieu sur un nouveau site mieux adapté : le site Estaing, ancien îlot industriel Michelin, situé entre les centres de Clermont-Ferrand et Montferrand. Site très bien adapté, car situé dans un emplacement à fort développement urbanistique et bien desservi, au carrefour de voies de communication importantes. Il améliorera de surcroît l'équilibre hospitalier entre le nord et le sud de l'agglomération clermontoise.

Le CHU Estaing (CHUE) comprend 565 lits (contre 537 à l'Hôtel-Dieu). Le bâtiment principal représente 68.000 m² de surface de plancher répartis sur quatre niveaux (sous-sol, rez-de-chaussée et deux étages). Une ligne spécifique du tramway passera au cœur même du site. D'importantes réflexions ont été menées par l'équipe d'architectes afin de concevoir un bâtiment adapté à des fonctions et à des hommes. Parallèlement, toute l'organisation des services de l'Hôtel-Dieu a été repensée. Le CHUE sera composé de treize services sous forme de huit pôles brièvement présentés ci-dessous :

- Le Pôle GORH sera constitué des services de consultation gynéco-obstétrique, de consultation génétique, de gynécologie, de chirurgie ambulatoire, de l'I.V.G., des grossesses à risque, mais aussi de l'assistance médicale à la procréation.
- Le Pôle Digestif avec la chirurgie digestive, l'hépatogastroentérologie, l'endoscopie, les consultations digestives, l'oncologie et la nutrition clinique.
- Le Pôle Imagerie comprendra la radio pédiatrie, l'échographie pédiatrique, l'échographie adulte, le scanner, l'IRM et l'imagerie interventionnelle.
- Le Pôle Administration Technique et Logistique regroupera des services comme l'atelier de maintenance, la pharmacie, la plateforme logistique, l'accueil ou encore la restauration.

- Le Pôle Spécialités Médico-chirurgicales regroupera les services de médecine interne, de dermatologie, d'hématologie clinique, des maladies infectieuses mais aussi la chirurgie maxillofaciale et les consultations médicochirurgicales.
- Le Pôle Pédiatrie regroupera la médecine pédiatrique, la chirurgie infantile, les urgences pédiatriques, la pédopsychiatrie, la réanimation néonatale, la réanimation pédiatrique, l'hématooncologie, ou encore, la dialyse pédiatrique.
- Le Pôle Anesthésie-Réanimation-Blocs avec le bloc opératoire, le bloc obstétrique, le bloc endoscopique, l'anesthésiologie, et la réanimation adulte.
- Le Pôle Laboratoire rassemblera les laboratoires de biochimie, d'hématologie, de fécondation in vitro, d'anatomie pathologie, et de biologie de la reproduction.

La Figure 7.9 donne les principaux services du NHE répartis sur les quatre niveaux dont dispose le CHUE.

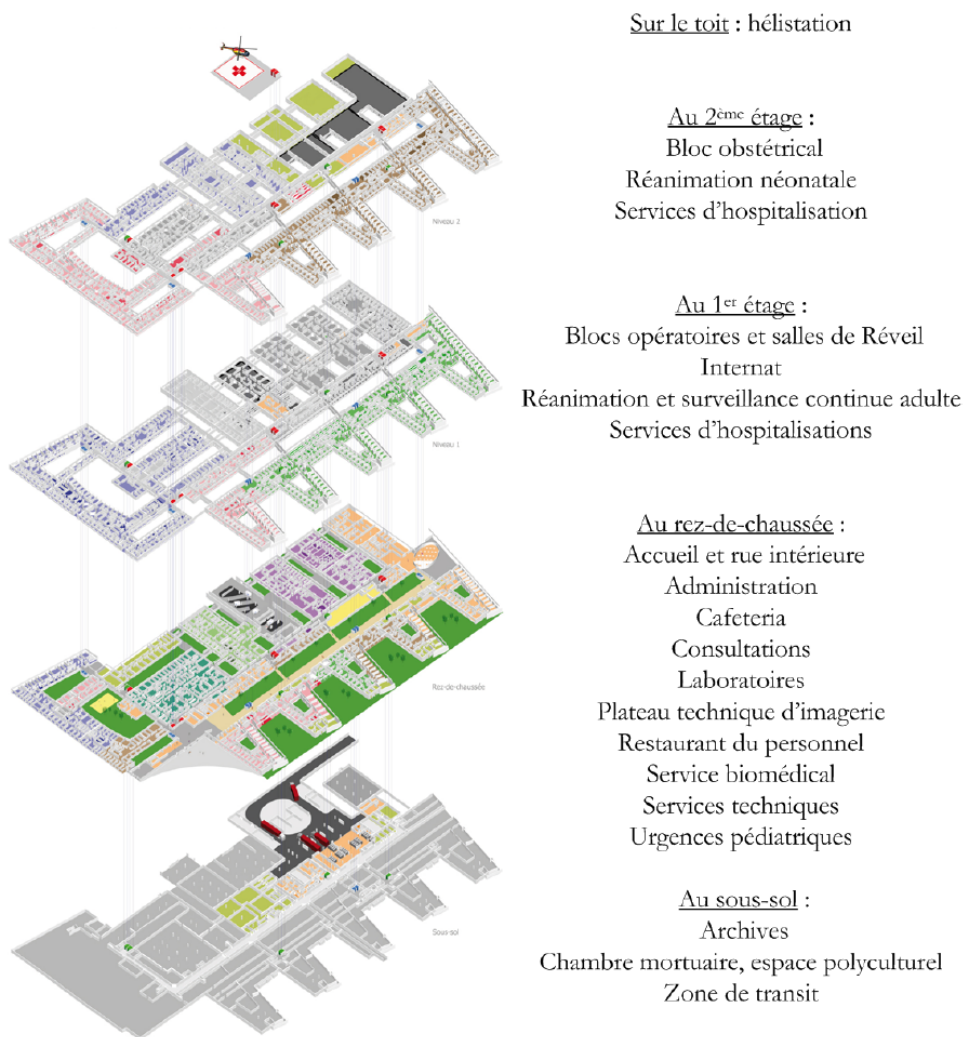


FIGURE 7.9 – Les différents niveaux du CHU Estaing [Rodier 10]

Au-delà de cette réorganisation fonctionnelle, le CHUE devra répondre aux exigences de simplicité, de modularité et de flexibilité. En effet, la construction d'un nouvel hôpital donne l'opportunité de repenser l'organisation hospitalière dans son ensemble et de redéfinir les missions du SI. C'est dans un contexte de modélisation de toutes les activités du CHUE et

de refonte de son SI que le CHU de Clermont-Ferrand s'est lancé dans le réingénierie de son CDM. L'arrivée à maturité, outre Atlantique, des logiciels et systèmes experts de prescription comme des robots d'ensachage de médicaments permettent d'envisager une refonte complète du circuit du médicament, basée sur une dispensation nominative journalière, intégrant les exigences en matière de traçabilité, d'opérationnalité et de performance économique.

7.4.3 Modélisation du CHU Estaing

Dans la perspective de l'ouverture du CHU Estaing qui a eu lieu en avril 2010, nous avons développé un modèle de simulation représentant le fonctionnement de cet hôpital avec une dispensation globale (fonctionnement prévu pour l'ouverture). Ce travail a été réalisé avec un étudiant en dernière année d'école d'ingénieur que nous avons encadré [Grimaldi 09].

7.4.3.1 La distribution

Nous avons à notre disposition plusieurs outils pour représenter un déplacement sous Automod. Parmi cela en figure deux intéressants : les *path mover system* (chariots filoguidés) et les *convoyors*.

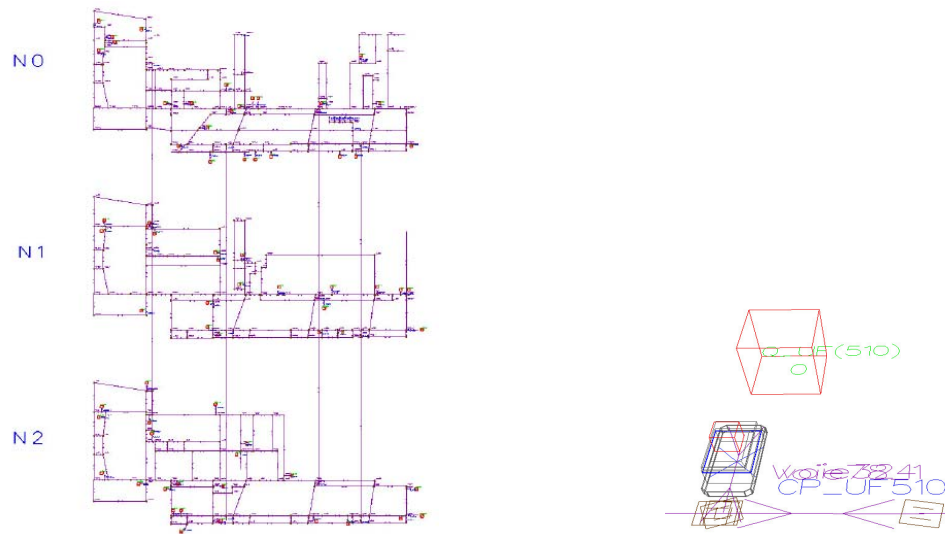
- Un *Path mover system* est un système de transport matériel dans lequel les véhicules ou les personnes se déplacent le long d'un chemin guidé, portant les entités (Loads) des emplacements de chargement aux emplacements de livraison.
- Un système de convoyeur consiste en une ou plusieurs sections. Les sections sont connectées l'une à l'autre par des transferts, qui adaptent la vitesse d'une charge et l'orientation quand la charge se déplace entre des sections. Les charges entrent dans un convoyeur à des points d'entrée et quittent ce convoyeur à des points appelés stations.

Étant donné la complexité des trajets à effectuer, nous avons choisi d'utiliser le *path mover system* pour une meilleure représentation. De plus, il est possible d'y créer des voies à double sens (ce qui se rapproche de la réalité). Sur la figure 7.10a), est représenté le circuit chariot et sur la figure 7.10b) est montrée la livraison d'une *Unité Fonctionnelle* (UF). Une UF correspond à une unité de soins (zone de stockage des médicaments) ; généralement un service est composé d'un UF, mais il peut en avoir plusieurs. On remarque que tous les chemins du CHUE sont représentés, ce qui rend ce modèle évolutif en cas de déplacement géographique d'UF.

Les chemins sont représentés à l'échelle. Nous fournissons en entrée une vitesse de déplacement. Les UF sont représentés par des *Queues* dans lequel un chariot va déposer des médicaments. A chaque UF correspond un *Control Point* (point de contrôle). Les points de contrôle sont des emplacements au niveau desquels les véhicules peuvent prendre ou déposer des charges dans le système. Les contrôles peuvent être placés n'importe où sur le circuit. L'envoi par chariot est effectué une fois par journée.

Les monte-charges sont représentés également par un *path mover*. Lorsque le chariot utilise le morceau de *path mover* délimitant le monte-charge, il utilise en même temps la ressource Monte-Charge correspondante. De plus, on placera des *blocks* pour limiter le trafic sur le chemin du monte-charge. Ainsi, un seul chariot à la fois peut utiliser le monte-charge (cela est modifiable en changeant la capacité du *block*).

Nous avons également choisi la solution du *path mover system* pour représenter le circuit pneumatique. On représente les 3 circuits par 3 *path mover systems* différents. Grâce à l'utilisation de *blocks* qui limitent le nombre de *vehicules* se déplaçant sur le même circuit, il ne peut y avoir qu'une seule cartouche se déplaçant en même temps dans chaque circuit (ce qui correspond à la réalité).



a) Circuit chariot dans le modèle de simulation du CHU Estaing b) Livraison d'une UF

FIGURE 7.10 – Livraison des UF du CHU Estaing

A chaque fois que l'on envoie des doses par pneumatique, on occupe la ressource cartouche. Le nombre de ressources est suffisamment grand pour que l'on ne soit pas limité en nombre d'envois de cartouches. Cela est conforme à la réalité car les cartouches vides seront renvoyées rapidement vers la pharmacie. Les médicaments sont envoyés dès que l'assemblage est réalisé par le préparateur. Il y a néanmoins une file d'attente à l'entrée du pneumatique au cas où une cartouche serait déjà dans le circuit. On envoie les médicaments destinés à être expédiés par pneumatique (consultation + estimations) dans une file d'attente (*queue*) provisoire en sortie de pneumatique, correspondant à l'UF. Ensuite, l'IDE ou le préparateur vient ranger le contenu de la cartouche dans l'armoire. On libère alors la ressource cartouche.

7.4.3.2 Rangement des UF

Le rangement des unités de soins est lancé par un événement créé à un horaire décidé à l'avance. Il ne s'effectue pas systématiquement à la suite de la distribution des médicaments dans les services. Le rangement peut être effectué par un IDE, par un préparateur ou par un magasinier. Ce choix se fait au début de la simulation à travers le fichier *Hypothese.txt*. On utilise des *order lists*, c'est-à-dire qu'à l'heure prévue de rangement, on range toutes les charges dans une file d'attente qui représente une armoire d'une UF. On utilise alors soit l'Agent Logistique, soit le préparateur, soit l'IDE, soit le magasinier.

7.4.3.3 La pharmacie du CHUE

A l'ouverture, le CHU était prévue pour fonctionner avec une dispensation globale, nous avons alors filtré le MGC et utilisé uniquement les entités correspondantes. Nous avons alors des préparateurs et des magasiniers qui s'occupent des *Doses Préparées pour un Service* (DPS) au niveau de la pharmacie. La zone de préparation est montré sur la figure 7.11.

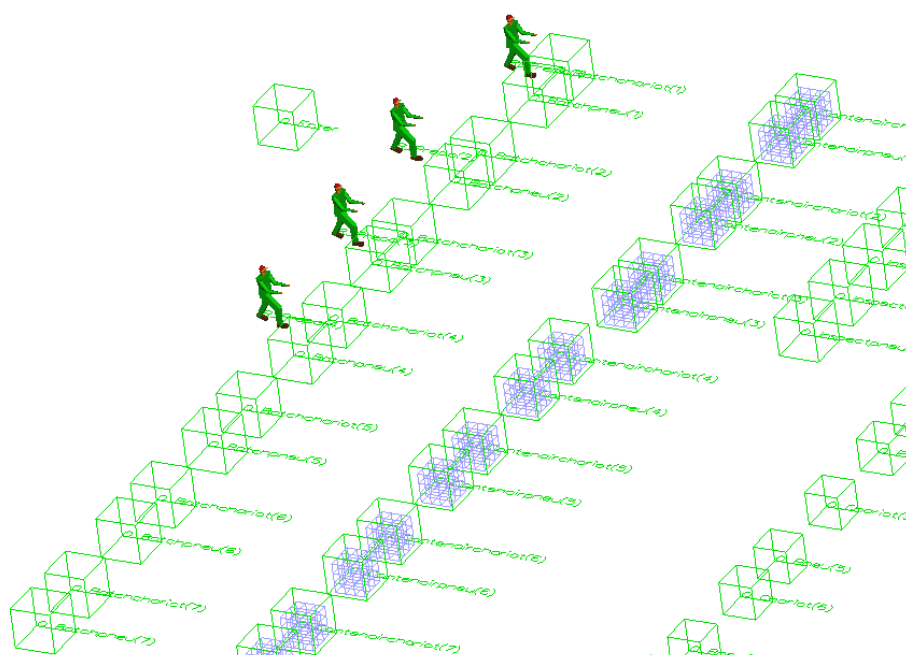


FIGURE 7.11 – Zone de préparation de la pharmacie globale

7.4.4 Interface d'entrée

7.4.4.1 Hypothèses de fonctionnement

L'interface d'entrée possède toutes les variables d'entrées nécessaires à l'évaluation des différentes hypothèses, c'est-à-dire qu'il comporte les différents paramètres modifiables. Il a été construit en suivant les diagrammes d'activité définis dans la section 6.3 pendant l'étape de Spécification en concordance avec le MGC du CDM. Ces diagrammes d'activité expliquent le comportement du CDM avec des diagrammes détaillant les activités préparation, livraison et rangement dans de nouveaux diagrammes. Les paramètres d'entrée correspondent au choix que doit faire l'utilisateur avant tout lancement du modèle de simulation. La modification de ces paramètres se fait grâce au fichier *Hypothese.txt*. Nous avons résumé ces hypothèses dans le tableau 7.5.

Hypothèses	Entrée	Choix
Source des données	1	PHARMA
	2	Statistiques
L'acteur qui effectue la livraison	1	Préparateur
	2	Agent Logistique
L'acteur qui effectue le rangement	1	Préparateur
	2	Agent Logistique
	3	IDE
Temps de montée du monte-charge		temps
Temps de préparation d'un caisson		temps
Temps de préparation d'un carton		temps
Temps de rangement d'UF		temps
Temps de chargement d'une cartouche		temps
Temps de préparation par ligne de médicament		temps
Temps de préparation par ligne de soluté		temps
Temps de rangement d'une cartouche		temps

TABLE 7.5 – Récapitulatif des hypothèses

A noter qu'une dispensation globale avec du plein/vide (KANBAN) sera représenté par des temps de préparation et de rangement plus faibles, et par des dotations de taille plus petite. Cela fera partie des variables d'entrée à modifier.

7.4.4.2 Listes des doses à préparer et évaluation du volume des médicaments transportés

L'interface d'entrée doit comporter aussi les listes des doses à préparer pour chaque service. Cela correspond à la liste des prescriptions pour une dispensation nominative ou à une liste de commandes des **UF** pour une dispensation globale ou globalisée. Nous avons deux façons possibles pour les mettre en entrée du modèle de simulation :

1. Par requête SQL, nous pouvons extraire du logiciel PHARMA le nombre de lignes de médicaments commandés pour chaque service et pour chaque patient pour le service de cardiologie. Le principal inconvénient est que comme nous ne connaissons pas le volume de chacun des 3800 médicaments se déplaçant dans l'hôpital, nous ne pouvons pas déterminer le volume exact total de médicaments transportés. Ce volume est pourtant nécessaire pour dimensionner et équilibrer en volume les tournées de livraison.
2. A partir des relevés du terrain où le nombre de colis livré par service a été compté sur deux semaines représentatives. L'avantage d'avoir le nombre de colis est de permettre d'avoir une idée du volume à transporter, et donc de savoir combien de chariots sont nécessaires pour une livraison. Un colis est soit un caisson, un carton, une glacière, un

bidon ou une bonbonne suivant ce qu'il contient. L'inconvénient est que l'on ne dispose pas de toutes les données liées au médicament. En effet, ces données ne fournissent que le nombre de colis se déplaçant dans le circuit, le type de chaque médicament transporté (soluté, gaz ou médicament) et l'UF dans laquelle doit être livré ce médicament. De ce fait, ces données seront inutilisables pour évaluer une dispensation nominative.

A partir des données fournies par PHARMA, nous pouvons évaluer le nombre de colis par UF à partir du nombre de lignes. Le nombre de chariots peut être évalué par le diagramme d'activité que nous avons présenté à la section 6.6 suivant les données en entrée. Un chariot est rempli directement par des médicaments (chariot à tiroirs) ou par des colis. Un chariot à tiroirs sera utilisé que dans le cas où la dispensation sera nominative. Si c'est ce type de chariot qui est utilisé, les données en colis ne devront pas être utilisées car elles ne peuvent l'être que pour un fonctionnement avec une livraison globale.

Les UF du CHUE ne seront pas les mêmes que celles de l'Hôtel Dieu. Certaines UF vont être créés, d'autres vont être supprimées, et d'autres vont être fusionnées. Pour évaluer les volumes de médicament par unité de soins, nous devons transposer au CHUE les données que l'on possède pour l'Hôtel-Dieu. Pour cela nous utilisons une matrice de correspondance HD-CHUE fournie par le service logistique du CHU.

7.4.4.3 Attribution des UF par tournée

Le choix des tournées dépend de la quantité et des volumes de médicaments distribués à chaque UF. Nous avons cherché à équilibrer de manière optimale cette répartition des médicaments. Pour résoudre ce problème, on étudie à l'avance les diverses tournées envisageables. Pour cela, nous nous basons sur les données relevées sur le terrain et récupérées sur PHARMA à l'Hôtel-Dieu pour les transposer en données prévisionnelles CHUE grâce à la matrice de transformation HD-CHUE. Les tournées ont alors été définies en collaboration avec le service logistique. Nous avons alors dressé une liste de tournées dans un fichier texte que l'on utilisera en entrée. Ces tournées seront paramétrables dans l'interface d'entrée. L'interface est constituée des fichiers *tournées1.txt*, *tournées2.txt*, *tournées3.txt*, *tournées4.txt* pour lire les données de chaque tournée.

7.4.5 Ajout de la pharmacie générique pour une livraison nominative

L'objectif de la réalisation d'une simulation d'une pharmacie nominative était non seulement de montrer l'intérêt de la dispensation nominative aux membres du CHU mais aussi de prendre en considération que cette simulation puisse être réutilisée dans d'autres hôpitaux.

L'objectif a alors été de réaliser une pharmacie dans un modèle propre qui pourrait être facilement implanté dans un modèle d'un hôpital fonctionnant sans pharmacie nominative et ainsi de pouvoir comparer les deux solutions (cf section 6.8.4.2). Il a fallu alors non seulement penser à la réalisation de cette simulation dans le cas du CHU de Clermont-Ferrand mais aussi pour tout autre cas, il a donc fallu faciliter au maximum l'interface entre la pharmacie et le reste de l'hôpital.

Quand on est dans un modèle, on peut en insérer un autre qui sera alors considéré comme un sous-modèle. Au niveau du code, la principale modification est l'ajout d'un attribut afin d'enregistrer la provenance des médicaments pour un service (de la pharmacie globale ou nominative). Les détails techniques du codage sont montrés en annexe A.4.

Dans le but de prévoir l'arrivée future de robots de dispensation, nous avons cherché, avec les pharmaciens, à mettre en place un scénario réaliste de dispensation nominative au

CHUE. Nous mettons en place la pharmacie générique correspondant aux caractéristiques du système que nous avons montré lors de la section 7.2.5, c'est-à-dire avec deux stations de travail équipées de robots et de deux stations de travail équipées de préparateurs. Après insertion de cette pharmacie, nous pourrions comparer différents types de fonctionnement.

Sur la figure 7.12, on peut voir la vue aérienne du modèle complet. Ce modèle est un assemblage de quatre composants différents :

1. pharmacie nominative
2. pharmacie globale
3. circuit pneumatique de livraison
4. circuit chariot de livraison

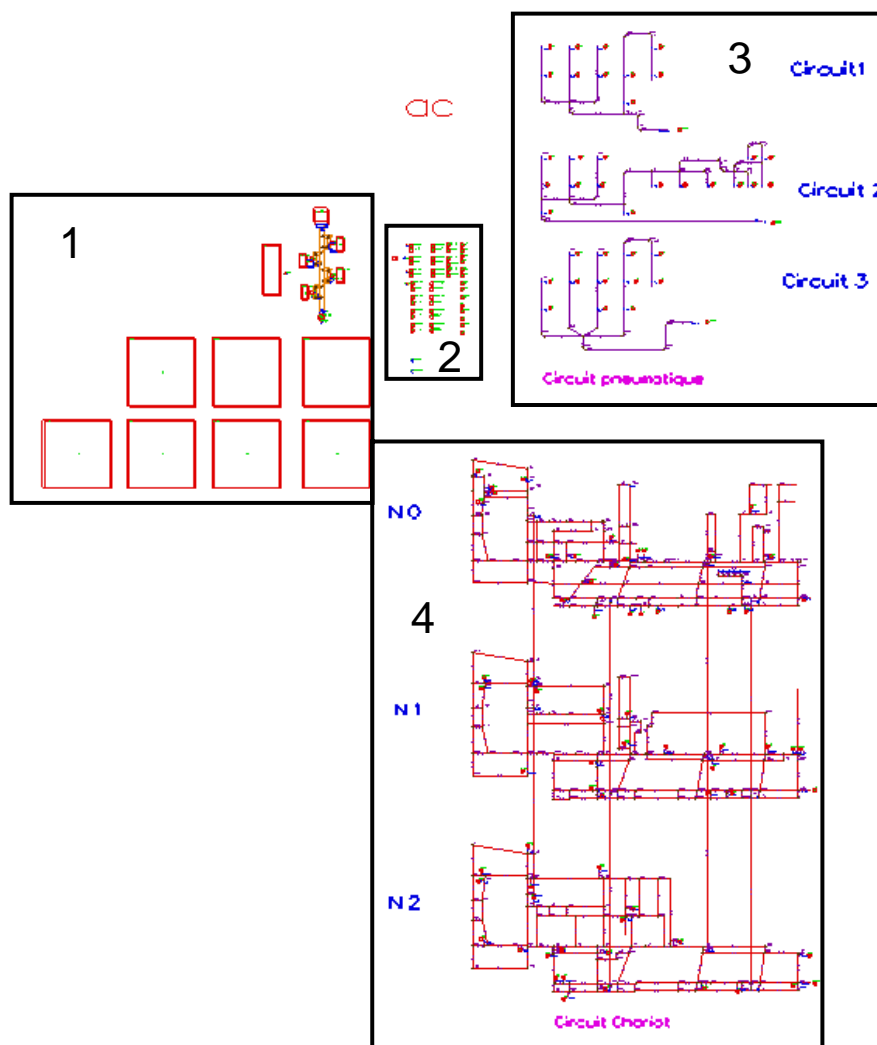


FIGURE 7.12 – Modèle complet avec une pharmacie avec une livraison nominative (1), une pharmacie avec une livraison globale (2), un circuit pneumatique (3) et un circuit chariot (4)

7.4.6 Bilan et exploitation du modèle d'action

7.4.6.1 Correspondance avec le MC et perspectives du modèle d'action

Ce modèle d'action est un exemple de synchronisation des flux physiques et informationnels. Le concept d'holon nous a permis de modéliser cette synchronisation. Un holon est une entité composée d'une partie physique (P_holon) et d'une partie informationnelle (I_holon). Ce modèle d'action a permis d'exploiter les holons et les interfaces du MC présentés à la section 7.3.

Le concept SCP est mis en œuvre par la pharmacie nominative. Le holon-produit communique avec les postes de travail équipés par les robots et gérés par les préparateurs. La liste des médicaments pour chaque poste de travail est donnée dans les fichiers *liste1.txt*, *liste2.txt*, *liste3.txt* et *liste4.txt*. Ces fichiers texte correspondent aux données qui seront transmises par le SI (le I_holon) de chaque robot et armoire. Le I_holon des postes de travail correspond aussi aux temps de cycle ajoutés au fichier *temps.txt*.

La lecture de la liste des prescriptions va permettre de créer « virtuellement » le holon-produit. Pour chaque holon-produit créé, une procédure va être lancée. Cette procédure va permettre de prendre une décision sur le parcours qui devra être effectué par le produit dans la pharmacie nominative. Elle fera évoluer les attributs de chaque produit en conséquence. La décision du chemin à parcourir est donc prise par le produit. Il est à noter que nous avons donc modélisé et simulé non seulement les flux physiques, mais aussi les flux d'informations. Le code et son explication sont proposés en annexe A.3.

Cette analyse pourrait être aussi améliorée en utilisant différents travaux réalisés sur la simulation des flux avec le concept SCP en rentrant dans le détail de l'ordonnancement de ce système d'assemblage. Cette communication correspond aux interfaces « assemblage automatique » et « assemblage manuel » que l'on retrouve sur le modèle représentant une dispensation nominative de la figure 7.8 présentée plus haut.

La synchronisation des flux physiques et informationnels s'effectue aussi avec l'exploitation des données basées sur PHARMA. Ces données représentant les prescriptions et les commandes des services correspondent au holon-ordre de notre MC. On pourrait aussi prendre directement les données utilisées dans CROSSWAY si l'on décide d'ajouter au modèle un module qui générerait la validation pharmaceutique. Cette exploitation des données correspond aux interfaces « prescription validée » des deux modèles.

Les interfaces « transmission par chariot » et « transmission par circuit automatisé » des deux modèles sont réalisées dans les fichiers correspondant aux ordonnances et aux commandes mises en entrée en ajoutant une variable pour chaque médicament afin de savoir s'il doit être livré par pneumatique ou par chariot. Le P_holon de chacun de ces composants correspond au circuit de livraison et leur I_holon, aux attributs qui leur sont donnés dans le logiciel de simulation.

Dans le modèle représentant une dispensation globalisée, il n'y a donc que les interfaces « manipulation par IDE » et « assemblage par IDE » qui ne sont pas modélisées. Elles pourraient être exploités en analysant le temps de travail de l'IDE : cela permettrait d'avoir une évaluation du gain de temps engendré par une dispensation nominative ou avec le système plein/vide (KANBAN). Dans le modèle représentant une dispensation nominative, l'unique interface qui n'est pas représentée est l'interface « validation ». Elle peut être exploitée en ajoutant un module de validation qui utiliserait les données des différents travaux qui ont eu lieu sur la simulation du CDM (cf section 6.8.1) ou en utilisant le temps mis par les pharmaciens pour valider actuellement le service de cardiologie.

Une autre perspective à ce modèle serait l'ajout en entrée des éléments stochastiques qui enverraient l'équivalent de commandes urgentes au niveau de la pharmacie.

7.4.6.2 Cadre expérimental

Pour exploiter ce modèle d'action dans le futur, nous avons mis en place un cadre expérimental qui permet de structurer toutes les études du modèle. Il résume l'objectif du modèle, les variables disponibles en entrée et en sortie, les hypothèses et la façon dont le modèle d'action alimente le modèle de résultats. Il devra être utilisé pour toute utilisation du modèle d'action. Ce modèle est déterministe.

- **Objectif** : Simuler un hôpital avec les deux types de pharmacie (globale et nominative) afin d'évaluer les performances de chacune.
- **Variables d'entrée** :
 - Pour la pharmacie générique :
 - Ordonnances (UF/patient/médicament/posologie/horaire)
 - Listes de médicaments par poste
 - Temps de picking poste
 - Pour le modèle du CHUE :
 - Commandes des services
 - Listes des hypothèses
 - Correspondance Station pneumatique / UF
 - Parcours de la livraison pneumatique et par chariot
- **Variables de sortie** :
 - Temps de préparation des médicaments
 - Utilisation des ressources (humains/robots)
- **Hypothèses** :
 - Vitesse de déplacement dans le circuit constant
 - Il n'y a pas de rupture de stock
- **Fin de la simulation** : au bout de 5 heures.
- **Récupération / Tri des données** : Les données alimentent le modèle de résultats qui exploite les données à l'aide du module Autostat d'Automod.

7.5 MODÈLE DE RÉSULTATS

7.5.1 Mise en œuvre du modèle d'action

Nous proposons de mettre en place un scénario que nous analyserons avec le logiciel Autostat. Ce module du logiciel Automod nous permet de lancer plusieurs itérations du modèle de simulation avec des paramètres d'entrée différents. Nous devons choisir les paramètres d'entrée que nous avons définis dans la table 7.5. Nous avons la possibilité dans Autostat de choisir quels paramètres changeront d'une itération à une autre. Nous décidons que le fichier d'hypothèses sera invariant et de faire varier d'une itération à une autre les données présentes dans les fichiers au format *.txt* correspondant aux prescriptions et aux demandes des services. La table 7.6 décrit les paramètres d'hypothèses choisis. Pour effectuer ce choix, nous pouvons travailler à partir des diagrammes d'activité du MGC. L'ensemble des hypothèses choisies correspondra à une sortie de chaque diagramme, c'est-à-dire une livraison de doses dans un service. Nous décidons que la livraison sera effectuée par un agent logistique, le rangement par un IDE, et les données seront celles extraites de la base de données de PHARMA. Les

pharmaciens évaluent le temps de préparation d'une ligne à 60s. Un monte-charge met environ 1min pour passer d'un étage à un autre (en comptant l'ouverture des portes). Nous évaluons le temps de rangement d'une cartouche à 1min dans le service. Les temps de préparation d'un caisson et d'un carton ne sont pas nécessaires car lors de l'utilisation des données issues de PHARMA, nous utilisons seulement les variables qui correspondent aux temps de préparation d'une ligne.

Hypothèses	Entrée	Choix
Source des données	1	Pharma
L'acteur qui effectue la livraison	2	Agent Logistique
L'acteur qui effectue le rangement	3	IDE
Temps de montée du monte-charge		60s
Temps de rangement d'UF		15min
Temps de chargement d'une cartouche		30s
Temps de préparation par ligne de médicament		60s
Temps de préparation par ligne de soluté		60s
Temps de rangement d'une cartouche		1 min

TABLE 7.6 – Choix des paramètres d'entrée

Il faut aussi remplir le fichier de *temps.txt* correspondant aux différents temps de la pharmacie générique (temps de préparation pour chaque station et temps d'entrée et sortie des boîtes sur le convoyeur). Nous utilisons un temps de préparation pour les robots de 10s par ligne de médicament. Ce chiffre est calculé à partir de la documentation du robot de Sinteco (robot que nous avons utilisé dans le MC). Elle est bien sûr approximative. Nous avons mis comme temps de préparation pour une ligne de médicament aussi bien pour les préparateurs que pour les magasiniers, 60s par ligne de commande (chiffre donné par les pharmaciens). Dans ce logiciel, on peut non seulement définir comme variable, un paramètre, une variable interne au programme mais aussi un fichier texte. En effet, dans notre cas, la plupart des variables sont sous forme de fichier texte, avec principalement les ordonnances et les demandes de services. On sélectionne les différents fichiers souhaités, et, lors des itérations, le fichier original sera automatiquement remplacé par un autre.

On réalise les itérations des mêmes ordonnances, dans un cas avec tous les services en dispensation globale, dans l'autre cas avec deux UF en dispensation nominative. Nous utilisons les données transmises du service cardiologie, ce service a deux UF différentes.

Le service de cardiologie est à l'hôpital Gabriel Montpied mais comme nous souhaitons que les données utilisées soient basées sur la réalité, nous décidons de « transférer virtuellement » ce service au CHUE. Le service est composé de deux *Unité Fonctionnelle* (UF), les UF 301 et 304. Les données lors de l'itération avec une dispensation nominative sont de deux types :

- Des données en nominative extraites du service de cardiologie de l'hôpital Gabriel Montpied. Nous avons simplifié les données non exploitables dans notre modèle (prescriptions qui ont été annulées...),

- Des données en globale extraites des services du CHUE. Nous avons retouché ces données pour pouvoir avoir une journée « moyenne ».

Concernant l'itération avec une dispensation globale, il n'y a plus que les données en globale extraites des services du CHUE auxquelles nous avons ajoutées les données nominatives précédentes « globalisées » et additionnées aux données globales.

7.5.2 Alimentation du modèle de résultats

Nous avons lancé les deux itérations avec les paramètres définis précédemment sous AutoStat. Nous avons étudié les taux d'occupation des ressources présentées sur la table 7.7.

R_{AL3}	Agent logistique qui effectue la livraison
R_{Robot1} et R_{Robot2}	Station de travail équipée d'un robot
$R_{Preparateur1}$ et $R_{Preparateur2}$	Station de travail avec un ajout manuel
$R_{Prepa(3)}$ et $R_{Prepa(4)}$	Préparateurs qui effectuent une préparation globale
$R_{Magasinier(1)}$ et $R_{Magasinier(2)}$	Magasiniers qui effectuent une préparation globale

TABLE 7.7 – Ressources représentées dans le modèle de résultats du scénario utilisé

Les deux graphiques suivants montrent le modèle de résultats. Le graphique de la figure 7.13 le représente dans le cas d'une dispensation nominative des deux UF du service de cardiologie, les autres UF fonctionnent avec une dispensation globale. Le graphique de la figure 7.14 le représente dans le cas du tout global avec les mêmes données que pour le premier, c'est-à-dire avec une globalisation des prescriptions ajoutée aux listes de médicaments commandés par les services.

Dans les deux graphiques, l'agent logistique (R_{AL3}) a le même taux d'utilisation. Ceci est logique, car la quantité de médicaments à livrer à chaque fois est la même. Les deux robots de préparation (R_{Robot1} et R_{Robot2}) et les préparateurs effectuant la préparation manuelle ($R_{Preparateur1}$ et $R_{Preparateur2}$) sont occupés lors d'une dispensation nominative. Ils ne sont logiquement plus utilisés lors d'une dispensation entièrement globale.

Dans le cas de la dispensation nominative, la grande majorité des médicaments sont traités par les robots, donc la charge de travail pour les préparateurs est très faible en comparaison du temps nécessaire en global. Le fait de faire passer les données nominatives dans la partie globale engendre une hausse de travail pour le préparateur chargé de préparer le pôle 3 ($R_{Prepa(3)}$) et pour le magasinier chargé de préparer les solutés pour le pôle 3 ($R_{Magasinier(1)}$). Le deuxième préparateur ($R_{Prepa(4)}$) et le deuxième magasinier ($R_{Magasinier(2)}$) représentés sur ces graphiques sont occupés au même taux car ils ne s'occupent que de la préparation des doses pour des pôles qui ont demandé une même quantité de médicaments et de solutés. Ce modèle de résultats permet d'évaluer les performances d'un scénario avant l'implantation dans le monde réel.

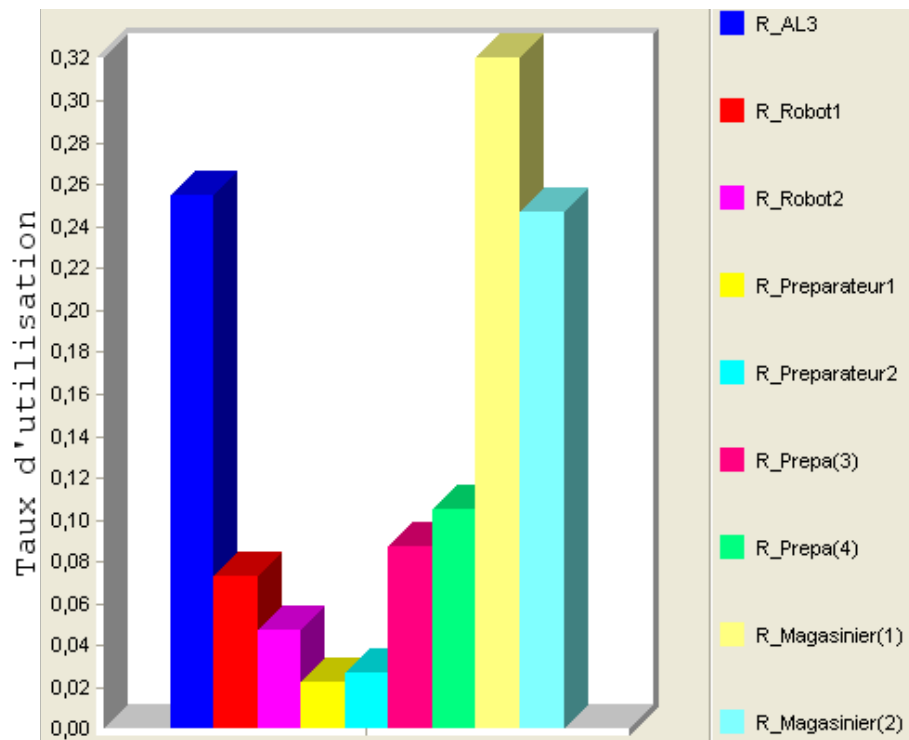


FIGURE 7.13 – Modèle de résultats pour un scénario avec une dispensation nominative

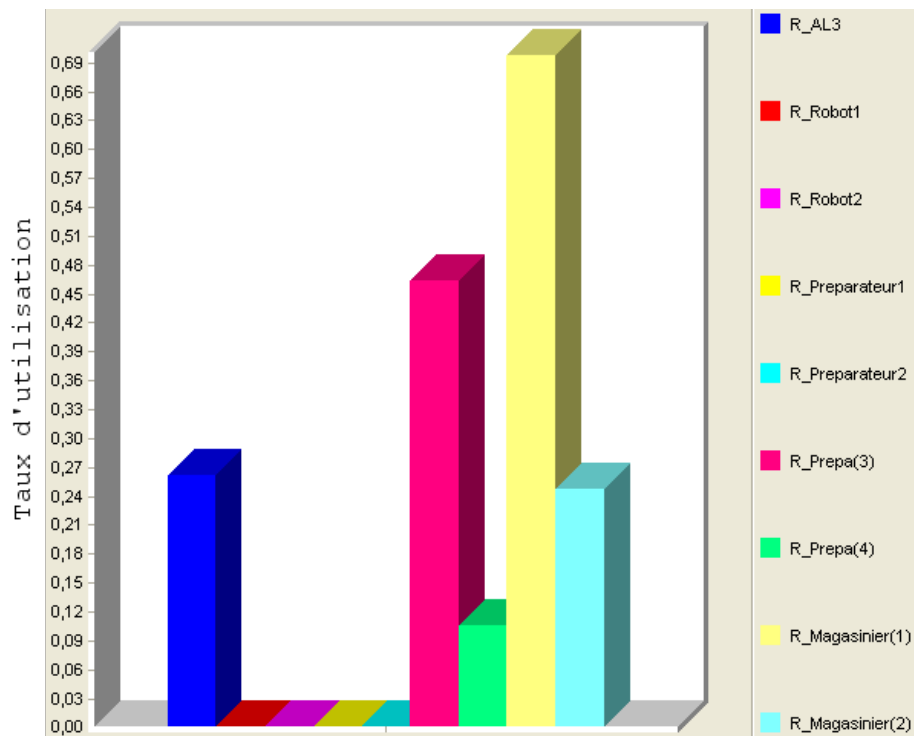


FIGURE 7.14 – Modèle de résultats pour un scénario avec une dispensation globale

7.6 CONCLUSION

Ce chapitre a présenté une instanciation du *Modèle Générique de Connaissance* (MGC) du *Circuit Du Médicament* (CDM) présentée au chapitre 6. Nous avons suivi les étapes préconisées par notre méthodologie ASCI-CDM pour permettre la modélisation d'un système. Ce système est le CHU de Clermont-Ferrand. Nous avons défini les caractéristiques de ce système en utilisant un formalisme très utilisé dans le milieu industriel. C'est un formalisme qui est aussi très facilement compréhensible pour les non-informaticiens, ce qui est un avantage non négligeable, car ces caractéristiques doivent être écrites avec des experts du milieu étudié (des pharmaciens pour le CDM). Il nous a permis de décrire de façon précise le comportement d'une pharmacie nominative en définissant les différents postes de travail et de stockage. Nous avons pu alors construire le *Modèle de Connaissance* (MC) du système étudié à partir du MGC du domaine. Nous avons instancié les éléments des diagrammes représentant les *Entités Décisionnelles* (ED) sur notre système. Nous avons ensuite élaboré le modèle d'action : nous avons repris et enrichi le modèle de simulation du CHU Estaing de Clermont-Ferrand qui avait été construit pour son ouverture. Nous l'avons enrichi en lui ajoutant une pharmacie générique permettant une dispensation nominative. Un modèle de résultats a été construit et est alimenté par ce modèle d'action. Ce modèle de résultats nous a permis de comparer les deux types de dispensation. Ces deux types de dispensation représentés dans le MGC par des diagrammes des ED qui ont été réalisés à partir d'une analyse d'un *Holonc Manufacturing System* (HMS) contrôlé par les ressources pour une dispensation globale et globalisée et à partir d'une analyse d'un HMS contrôlé par le produit pour la dispensation nominative.

La méthodologie que nous avons développée nous a permis de construire un outil d'aide à la décision qui permettra aux pharmaciens et managers du CHU de Clermont-Ferrand de mieux réorganiser le CDM lors de la mise en place de la dispensation nominative robotisée ou non.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

CONCLUSION GÉNÉRALE

Dans le **chapitre 1**, nous avons exposé les problématiques liées au *Circuit Du Médicament (CDM)* dans un hôpital. Le principal avantage recherché est la réduction des erreurs médicamenteuses. Le **CDM** se décompose en trois étapes : prescription du médecin, dispensation par les pharmaciens et leurs préparateurs, et administration par l'infirmier. Nous avons exposé les différents types de dispensation et nous avons mis en avant la dispensation nominative (préparation des doses pour chaque patient avant livraison) par rapport à une dispensation globale (livraison à partir de la commande des services) et globalisée (addition des prescriptions à la pharmacie et livraison par service). La dispensation nominative permet une meilleure sécurité mais nécessite l'utilisation de nouvelles technologies. Le plus important des résultats attendus est une meilleure exécution des soins aux patients et un meilleur contrôle par une meilleure traçabilité dans la préparation des médicaments en utilisant les technologies d'identification telles que le code à barre ou la **RFID**. En collaboration avec les pharmaciens du **CHU** de Clermont-Ferrand, nous avons alors proposé d'équiper les *Doses Préparées pour un Patient (DPP)* à la pharmacie lors d'une dispensation nominative avec une technologie d'identification. Nous avons remarqué dans ce cas que le **CDM** est comparable à un *Flexible Manufacturing System - Système manufacturier flexible (FMS)* où l'ajout d'un médicament correspond à un assemblage sur un système manufacturier. Des travaux récents sur les **FMS** ont proposé de piloter ces systèmes avec le produit en proposant le concept de *Système Contrôlé par le Produit (SCP)*. Nous avons montré que nous pouvons appliquer sur le **CDM** les connaissances et les recherches sur ce concept. Il utilise le paradigme de *Holonic Manufacturing System (HMS)* pour permettre une synchronisation des flux physiques et informationnels. Un holon est une entité ayant une partie informationnelle et une partie physique et pouvant se décomposer en autres holons.

Dans le **chapitre 2**, nous avons détaillé le concept **SCP** qui est utilisé actuellement pour mettre de l'intelligence dans les produits afin que ceux-ci puissent, en temps réel, décider de leur évolution (transport, gamme de fabrication...) dans le but d'améliorer la flexibilité, la modularité et la réactivité dans un processus de production. Nous avons aussi constaté que le *Système d'Information Hospitalier (SIH)* peut être comparé à un *Système d'Information (SI)* d'une entreprise. Nous avons exposé des travaux récents utilisant la notion de *Système de Systèmes (SdS)* pour permettre une meilleure interopérabilité entre différents **SI**. Dans ce contexte, le produit peut être considéré comme le pivot devant assurer la cohérence entre le niveau de planification (**ERP**) et d'exécution de la fabrication (**MES**) [Cardin 07].

Dans le **chapitre 3**, nous avons présenté les cadres d'architectures, les méthodologies, les méthodes et les langages les plus usités pour modéliser et concevoir un système de production. Nous avons remarqué des travaux intéressants utilisant l'approche par composant. Un système est alors vu comme un assemblage de composants dans un certain contexte, chaque composant pouvant être décomposé en autres composants de niveau d'abstraction in-

férieur. Des travaux proposent une modélisation de chaque composant par les langages **UML** et **SysML** [OMG 10b, OMG 10a] pour en décrire le comportement statique et dynamique tel que la méthodologie **MBCSA** [Chiron 09]. La démarche **MDA** qui propose une architecture pour passer d'un niveau d'abstraction à un autre nous a parue aussi intéressante. La méthodologie **ASCI** qui a déjà été utilisée dans le milieu hospitalier a aussi retenu notre attention.

Dans le **chapitre 4**, nous avons étudié les architectures, méthodologies et langages utilisés pour modéliser et concevoir un **HMS** contrôlé par les ressources et un **HMS** contrôlé par le produit en reprenant des outils présentés au chapitre 3.

A partir de cet état de l'art, dans le **chapitre 5**, nous avons développé une méthodologie complète de réingénierie de **HMS** : la *Méthodologie de Réingénierie des HMS* (**MRHMS**). Nous avons décrit notre vision d'une entreprise dans un méta-méta-modèle qui permet de définir la structure d'une entreprise et les interactions entre chaque composant. Puis nous avons spécialisé le méta-méta-modèle avec deux méta-modèles à un niveau d'abstraction inférieur qui représentent un **HMS** contrôlé par les ressources et un **HMS** contrôlé par le produit. Nos méta-modèles ont alors une généricité relative aux **HMS**. Le passage d'un méta-modèle à un autre représente l'acte de réingénierie. Pour faire le lien entre ces méta-modèles, un domaine (une classe de **HMS**) et un système du domaine (l'univers du discours), nous avons recherché une méthodologie qui s'intègre dans **MRHMS**.

En effet, nous avons regardé les propositions méthodologiques pouvant structurer ces concepts et architectures afin de répondre à notre problématique de réingénierie pour le contrôle par le produit. Nous nous sommes arrêté sur deux travaux :

- la *Méthodologie Bidirectionnelle de Conception des Systèmes Automatisés* (**MBCSA**) qui utilise une architecture **MDA** ainsi que l'approche par composant pour la conception de systèmes automatisés,
- la méthodologie *Analyse, Spécification, Conception, Implémentation* (**ASCI**) qui propose l'étude d'un domaine en préconisant la construction d'un *Modèle Générique de Connaissance* (**MGC**) ainsi qu'une bibliothèque de composants réutilisables. **ASCI** a déjà été utilisé dans le domaine du milieu hospitalier [Rodier 10] et pour la conception de *Système Multi-Agent* (**SMA**) [Ali 09]. Nous avons fait le constat que le concept de **SMA** est proche de celui de **HMS**.

Nous avons démontré que ces deux méthodologies étaient proches et se complétaient, nous avons alors proposé une méthodologie **ASCI** enrichie par **MBCSA** et deux diagrammes représentant les *Entités Décisionnelles* (**ED**) comme proposé dans [El Haouzi 08b]. Nous l'avons appelé *ASCI pour la Réingénierie des HMS* (**ASCI-RHMS**). Dans notre méthodologie, nous avons utilisé les langages **UML** et **SysML**. Nous avons expliqué que le langage **UML** est adapté pour modéliser la décomposition systémique héritée du **ASCI** standard et que **SysML** est plus précis pour modéliser les **ED**. En effet, nous avons utilisé ce langage, préconisé par **MBCSA**, car il nous permet de formaliser facilement l'approche par composant. Ce langage est adapté à notre propos grâce à la notion de « port » et d'« interface » et à la décomposition par niveau d'abstraction qu'il permet. Chaque *Internal Block Diagram - diagramme interne de bloc* (**IBD**) de **SysML** peut se décomposer en autres **IBD** et à chaque niveau d'abstraction, on peut assembler des composants dans des contextes différents avec un **IBD** par contexte. On peut alors étudier différents scénarios ou différentes implémentations sur différentes plateformes de façons cohérentes.

pour résumer, notre méthodologie **MRHMS** peut être utilisée de trois façons :

1. En partant des méta-modèles représentant la réingénierie, on peut construire un **MGC** d'un domaine puis le *Modèle de Connaissance* (**MC**) d'un système ; il s'agit alors d'une étude complète d'un système et de son domaine ;

2. En partant des méta-modèles, on peut uniquement construire le **MGC** d'un domaine à étudier ; il s'agit de l'étude d'une classe de **HMS** (ce que nous avons fait au chapitre 6) ;
3. En partant du **MGC** d'un domaine, on peut construire le **MC** d'un système de ce domaine ; il s'agit de l'étude d'un système d'un domaine qui a déjà été préalablement étudié (ce que nous avons fait au chapitre 7).

Nous avons alors appliqué la méthodologie **MRHMS** au domaine du **CDM** dans le **chapitre 6**. Nous avons construit le **MGC** du **CDM** via les étapes d'Analyse et de Spécification tel que préconisé par **ASCI-RHMS**. Nous avons montré quelques cas d'étude de Conception et d'Implémentation de composants ajoutés dans la bibliothèque de composants réutilisables :

- La conception d'une interface d'entrée/sortie pour logiciel de simulation,
- Une étude de l'interface entre le **SIH** et les *Doses Préparées pour un Patient* (**DPP**) pour permettre une synchronisation des flux physiques et informationnels,
- La conception de deux modèles de simulation : une pharmacie générique avec une dispensation nominative et une avec une dispensation globale. Ces pharmacies génériques pouvant être adaptables à n'importe quel établissement de santé en France.

Dans le **chapitre 7**, nous avons instancié le **MGC** à un système particulier : le **CHU** de Clermont-Ferrand. Nous avons étudié un scénario de dispensation nominative en utilisant un formalisme très utilisé dans le monde industriel : la norme JIS Z 8206. Pour construire le **MC**, nous avons instancié les diagrammes des **ED** du **MGC**. Un modèle d'action, élaboré à partir du **MC** et basé sur le logiciel de simulation AUTOMOD, nous permet de comparer un fonctionnement avec une dispensation nominative et une dispensation globale en ajoutant un composant développé dans la bibliothèque de composant : un modèle de pharmacie générique fonctionnant avec une dispensation nominative. Ce modèle simule le **CDM** en utilisant les données des différents **SI**. En d'autres termes, nous avons simulé non seulement des flux physiques mais aussi les flux d'information via le concept d'holon : nous avons alors une synchronisation des flux physiques et informationnels. Nous avons contrôlé la pharmacie nominative en mettant en œuvre le concept de **SCP**. Nous pouvons alors évaluer les performances de différents fonctionnements et proposer un outil d'aide à la décision pour les pharmaciens et les managers du **CHU**. Nous avons montré un scénario avec une évaluation de performances utilisant cet outil.

Nous avons résumé notre approche méthodologique et ces applications dans ce manuscrit dans la table 7.8.

M ₃	Méta-Méta-Modèle	Définition d'un HMS	Chapitre 5
M ₂	Méta-Modèle	Modélisation d'une classe de HMS	Chapitre 5
M ₁	Modèle	Modélisation d'un CDM	Chapitre 6
M ₀	Univers du discours	CHU de Clermont-Ferrand	Chapitre 7

TABLE 7.8 – Approche développée dans cette thèse

Cette méthodologie permet plus aisément et de façon cohérente, de modéliser, simuler et déployer une réorganisation d'un **CDM** et plus généralement de n'importe quel **HMS**. Notre objectif a été de construire une bibliothèque de composants réutilisables pour n'importe quel **CDM**. A partir de cette bibliothèque, nous avons développé un modèle de simulation pour élaborer un outil d'aide à la décision. Tous les résultats pourront être utilisés pour réorganiser

le **CDM** au **CHU** de Clermont-Ferrand prévu dans quelques années et ont été utilisés pour l'organisation d'un nouvel hôpital qui a ouvert récemment.

PERSPECTIVES

Perspectives générales

Une perspective serait le développement d'une méthodologie pour la transformation d'un modèle représentant un contrôle par les ressources vers un modèle représentant un contrôle par le produit. Une piste serait l'utilisation de méthodologies tels que **MaSE** (cf section 3.3.4.1) ou **TROPOS** (cf section 3.3.4.2) ou encore celle présentée dans [Kisti 05].

Les modèles pourraient aussi être améliorés en prenant en compte le fait que les entreprises et les hôpitaux peuvent avoir plusieurs sites, nous pourrions alors gérer l'organisation et le transport de produits ou de doses de médicaments entre les différents établissements [Augusto 09, Azevedo 00].

Une autre perspective de cette thèse serait d'étudier le **CDM** d'un autre point de vue que celui développé dans ce manuscrit, comme par exemple par l'approche par processus métier **BPM** [Cauvet 08]. Cette approche fera l'objet d'une thèse prochainement.

Perspectives pour les Méta-modèles (niveau M2)

A partir des méta-modèles représentant un système contrôlé par les ressources et un système contrôlé par le produit, une perspective serait de les instancier à la réingénierie d'autres classes de **HMS** pour vérifier son applicabilité. Concrètement, il faudra étudier un **HMS** autre que le **CDM**, c'est-à-dire un domaine concerné par la personnalisation de masse (ex : le domaine de la production automobile). On pourrait aussi construire des méta-modèles centrés sur le **SI** ou d'autres éléments du méta-méta-modèle pour permettre une réingénierie autre que celle d'un atelier de production.

Perspectives pour le Modèle Générique du circuit du médicament (niveau M1)

A partir du **MGC** construit dans cette thèse, une perspective serait de l'appliquer à d'autres hôpitaux français qui cherchent à moderniser leur **CDM**. Nous pouvons aussi envisager d'écrire d'autres composants génériques pour des modèles de simulation en prenant en compte la validation des prescriptions par les pharmaciens. Nous pouvons aussi ajouter des éléments stochastiques comme par exemple la gestion des urgences des services en contrôlant l'arrivée des prescriptions et des commandes dans le modèle.

La conception de composant pour différentes implémentations, comme par exemple sur d'autres logiciels de simulation, serait une perspective intéressante pour construire une bibliothèque de composants réutilisables la plus riche possible pour les établissements de santé. Une perspective intéressante serait la confrontation des modèles développés dans ce manuscrit avec la législation de différents pays pour rendre ce modèle réutilisable d'un point de vue international. Une autre perspective serait la décomposition des composants à un niveau encore plus fin comme par exemple rentrer dans le détail du pilotage par le produit avec des langages comme les *Réseaux de Petri* (**RdP**) comme le proposent plusieurs auteurs [Leitão 04, Demongodin 08]. Cette décomposition pourrait être implantée pour améliorer le modèle de simulation que nous avons développé.

Perspectives au CHU de Clermont-Ferrand (niveau Mo)

La perspective que l'on peut citer au niveau Mo est l'application de cette méthodologie pour le dimensionnement et la réingénierie de la pharmacie centrale de Clermont-Ferrand avec l'outil d'aide à la décision élaboré à partir de notre méthodologie. Nous pourrions ainsi évaluer les performances de différentes organisations sur les différents sites du **CHU**.

Nous disposons maintenant d'un cadre méthodologique permettant de réorganiser le **CDM** d'un hôpital en France. Dans un contexte de sécurisation et de réduction budgétaire, nous participons ainsi au mouvement de restructuration des hôpitaux français pour mieux répondre aux attentes des patients et de l'état .

BIBLIOGRAPHIE

- [Ali 09] Ahmad Ali. *L'approche multi-agents pour le pilotage des systèmes complexes appliquée aux Systèmes du Trafic Urbain*. Thèse de doctorat, Université Blaise Pascal - Clermont II, 2009.
- [Aloui 07] Saber Aloui. *Contribution à la modélisation et l'analyse du risque dans une organisation de santé au moyen d'une approche système*. Thèse de doctorat, Ecole des Mines de Paris, Paris, France, 2007.
- [AMICE 93] ESPRIT Consortium AMICE, editeur. *Cimosa : Open system architecture for cim (research reports esprit / project 688/5288. amice)*. Numéro ISBN 3-540-56256-7. Springer ; 2nd, rev. and ext. ed. edition, july 1993.
- [André 10] Virginie André, Nathalie Grangeon, Sylvie Norre & Frédéric Philippe. *Un problème de transport dans un environnement hospitalier modélisé par un PDP-TW à contraintes de ressources*. In GISEH'10 : Conférence Francophone Gestion et Ingénierie des Systèmes Hospitaliers, 2010.
- [Arriudarre 98] Arriudarre, Taillefer, Rivière & Landi. *Intérêt de l'automatisation de la dispensation nominative journalière des médicaments par l'ATC212 : réflexion au CHU d'Aix-en-Provence*. ADPHRO, vol. tome 23, pages 41–46, 1998.
- [Augusto 09] Vincent Augusto & Xiaolan Xie. *Redesigning pharmacy delivery processes of a health care complex*. Health Care Management Science, vol. 12, pages 166–178, 2009. 10.1007/s10729-008-9086-3.
- [Auzelle 08] Jean-Philippe Auzelle, Gérard Morel, Hervé Panetto & Frédéric Mayer. *Using Systems of Systems Engineering to Improve the Integration of Enterprise-Control Systems*. Special issue on Systems Engineering : Best of France Insight journal of INCOSE, vol. 11/3, 2008.
- [Auzelle 09] Jean-Philippe Auzelle. *Proposition d'un cadre de modélisation multiéchelles d'un système d'information en entreprise centré sur le produit*. Thèse de doctorat, Université Henri Poincaré, Nancy I, mars 2009.
- [AWG 97] C4ISR AWG. *C4ISR Architecture Framework Version 2.0*. Rapport technique, Departement of Defense USA, février 1997.
- [Azevedo 00] A.L. Azevedo & J.P. Sousa. *A component-based approach to support order planning in a distributed manufacturing enterprise*. Journal of Materials Processing Technology, vol. 107, pages 431–438, 2000.
- [Baïna 06a] Salah Baïna. *Interopérabilité dirigée par les modèles : Une Approche Orientée Produit pour l'interopérabilité des systèmes d'entreprise*. PhD thesis, Université Henri Poincaré, Nancy I, décembre 2006.
- [Baïna 06b] Salah Baïna & Gérard Morel. *Product Centric Holons For Synchronisation And Interoperability In Manufacturing Environments*. In The 12th IFAC Symposium on Information Control Problems in Manufacturing, pages 431–438, St Etienne, France, 2006.

- [Barbier 02] Franck Barbier, Corine Cauvet, Mourad Oussalah, Dominique Rieu, Sondes Bennisri & Carine Souveyet. *Composants dans l'ingénierie des systèmes d'information : concepts clés et techniques de réutilisation*. In Actes des deuxièmes assises nationales du GdR I3, 2002.
- [Barkmeyer 99] Contributors Ed Barkmeyer, Peter Denno, Shaw Feng, Al Jones & K. Wallace. *NIST Response to MES Request for Information*, 1999.
- [Bauer 01] Bernhard Bauer, Jörg Müller & James Odell. *Agent UML : A Formalism for Specifying Multiagent Software Systems*. Agent-Oriented Software Engineering, pages 109–120, 2001.
- [Bedouch 09] P. Bedouch, M. Baudrant, M. Detavernier, C. Rey, É. Brudieu, L. Foroni, B. Allenet & J. Calop. *La sécurisation du circuit du médicament dans les établissements de santé : données actuelles et expérience du centre hospitalier universitaire de Grenoble*. Annales Pharmaceutiques Françaises, vol. 67, no. 1, pages 3 – 15, 2009.
- [Bézivin 01] Jean Bézivin. *Towards a precise definition of the OMG/MDA framework*. In ASE, pages 273–280. IEEE Computer Society, 2001.
- [Blanc 08] Pascal Blanc, Isabel Demongodin & Pierre Castagna. *A holonic approach for manufacturing execution system design : An industrial application*. Engineering Applications of Artificial Intelligence, vol. 21, no. 3, pages 315 – 330, 2008.
- [Blobel 08] Bernd Blobel. *Application of the component paradigm for analysis and design of advanced health system architectures*. International Journal of Medical Informatics, vol. 60, pages 281–301, 2008.
- [Boardman 06] John Boardman & Brian Sauser. *System of systems : The meaning of of*. In IEEE/SMC International Conference on System of Systems Engineering, Los Angeles, USA, 2006.
- [Bolognesi 87] Tommaso Bolognesi & Ed Brinksma. *Introduction to the ISO specification Language LOTOS*. Computer Networks and ISDN Systems, vol. 14, no. 1, 1987.
- [Bongaerts 00] Luc Bongaerts, László Monostori, Duncan McFarlane & Botond Kádár. *Hierarchy in distributed shop floor control*. Computers in Industry, vol. 43, no. 2, pages 123 – 137, 2000.
- [Bourrières 90] J.-P. Bourrières. *Contribution à la modélisation intégrée des systèmes robotisés d'assemblage*. PhD thesis, UFR de sciences et techniques de l'Université de Franche-Comté, Besançon, 1990.
- [Boussier 07] Jean-Marie Boussier. *Modélisation de comportements dans les systèmes dynamiques : Application à l'organisation et à la régulation de stationnement et de déplacement dans les Systèmes de Trafic Urbain*. Thèse de doctorat, Université de La Rochelle, La Rochelle, France, 2007.
- [Bresciani 04] Paolo Bresciani, Anna Perini, Paolo Giorgini, Fausto Giunchiglia & John Mylopoulos. *Tropos : An Agent-Oriented Software Development Methodology*. Autonomous Agents and Multi-Agent Systems, vol. 8, no. 3, pages 203–236, May 2004.
- [Brown 96] Alan W. Brown & Kurt C. Wallnau. *Engineering of Component-Based Systems*. In 2nd International Conference on Engineering of Complex Computer Systems, volume tome 23, pages 414–422. IEEE Computer Society Press, 1996.

- [Brussel 98] Hendrik Van Brussel, Jo Wyns, Paul Valckenaers, Luc Bongaerts & Patrick Peeters. *Reference architecture for holonic manufacturing systems : PROSA*. Computers in Industry, vol. 37, no. 3, pages 255 – 274, 1998.
- [Bézivin 05] Jean Bézivin. *On the unification power of models*. Software and Systems Modeling, vol. 4, no. 2, pages 171–188, mai 2005.
- [Cairns 01] Chris Cairns. *Robots and automation : a UK perspective*. Hospital Prescriber Europe, vol. spring 2002, pages 33–34, 2001.
- [Calvez 90] Jean-Paul Calvez. *Spécification et conception des systèmes : une méthodologie mcse*, volume ISBN : 2-225-82107-0. Normandie Impr., 1990.
- [Cardin 07] Olivier Cardin. *Apport de la simulation en ligne dans l'aide à la décision pour le pilotage des systèmes de production : Application à un système flexible de production*. Thèse de doctorat, Université de Nantes, 2007.
- [Cardin 08] Olivier Cardin, Pierre Castagna & Étienne Chové. *La simulation pour l'aide au pilotage des systèmes contrôlés par le produit*. In 7e Conférence Internationale de MODélisation et SIMulation, Paris, France, 2008.
- [Cauterman 07] Maxime Cauterman, Hervé Laborie & Sébastien Woynar. *Circuit du médicament Approfondissement Rapport intermédiaire*. Rapport technique, mission nationale d'expertise et d'audit hospitaliers, 2007.
- [Cauvet 08] Corine Cauvet & Gwladys Guzelian. *Business Process Modeling : A Service-Oriented Approach*. In HICSS '08 : 41st Annual Hawaii International Conference on System Sciences, page 98, Washington, DC, USA, 2008. IEEE Computer Society.
- [Cea Ramirez 06] Aldo Alexis Cea Ramirez. *Contribution à la Modélisation et à la Gestion des Interactions Produit-Processus dans la Chaîne Logistique par l'Approche Produits Communicants*. Thèse de doctorat, Université Henri Poincaré, Nancy-I, 2006.
- [Cernuzzi 06] Luca Cernuzzi & Franco Zambonelli. *Dealing with adaptive multi-agent organizations in the gaia methodology*, volume 3950/2006, pages 109–123. Août 2006.
- [Chauvet 09] Julie Chauvet. *Une méthodologie de modélisation pour les systèmes hospitaliers : application sur le Nouvel Hôpital d'Estaing*. Thèse de doctorat, Université Blaise Pascal - Clermont II, Clermont-Ferrand, France, 2009.
- [Chikofsky 90] Elliot J. Chikofsky & James H. Cross II. *Reverse Engineering and Design Recovery : A Taxonomy*. IEEE Software, vol. 7, pages 13–17, 1990.
- [Chirn 00] Jin-Lung Chirn & D. C. Mcfarlane. *A holonic component-based approach to reconfigurable manufacturing control architecture*. pages 219–223, 2000.
- [Chiron 07] Fabien Chiron & Khalid Kouiss. *Design of IEC 61131-3 Function Blocks using SysML*. In the 15th Mediterranean Conference on Control & Automation, Athènes, Grèce, juillet 2007.
- [Chiron 09] Fabien Chiron. *Contribution à la flexibilité et à la rapidité de conception des systèmes automatisés avec l'utilisation d'UML*. Thèse de doctorat, Université Blaise Pascal - Clermont II, Clermont-Ferrand, France, 2009.
- [Combes 94] Catherine Combes. *Un environnement de modélisation pour les systèmes hospitaliers*. Thèse de doctorat, Université Blaise Pascal - Clermont II, Clermont-Ferrand, France, 1994.

- [Courtiat 00] J. P. Courtiat, C. A. S. Santos, C. Lohr & B. Outtaj. *Experience with RT-LOTOS, a temporal extension of the LOTOS formal description technique*. Computer Communications, vol. 23, no. 12, pages 1104 – 1123, 2000.
- [Darmoul 10] Saber Darmoul. *Étude de la contribution des systèmes immunitaires artificiels au pilotage de systèmes de production en environnement perturbé*. Thèse de doctorat, Université Blaise Pascal, Clermont II, Clermont-Ferrand, France, 2010.
- [David 92] René David & Hassane Alla. *Du grafcet aux réseaux de petri*, volume ISBN 2-86601-325-5. Paris, 2e édition, 1992.
- [Deen 03] S. M. Deen. *Agent based manufacturing : advances in holonic approach*. New York : Springer, 2003.
- [Deloach 01] Scott A. Deloach, Mark F. Wood & Clint H. Sparkman. *Multiagent Systems Engineering*. International Journal of Software Engineering and Knowledge Engineering, vol. 11, no. 3, pages 231–258, 2001.
- [Demongodin 08] Isabel Demongodin & Jean-Claude Hennet. *A Petri net model of distributed control in a holonic Manufacturing Execution System*. In the 17th World Congress The International Federation of Automatic Control, pages 15939–15944, Seoul, Korea, Juillet 2008.
- [DHOS 01] DHOS. *Informatisation du circuit du médicament dans les établissements de santé*. ministère de la santé, France, direction de l’hospitalisation et de l’organisation des soins édition, 2001.
- [Di Martinelly 08] Christine Di Martinelly. *Proposition of a framework to reengineer and evaluate the hospital supply chain*. Thèse de doctorat, Institut National des Sciences Appliquées de Lyon et Facultés Universitaires Catholiques de Mons, Lyon, France et Mons, Belgique, 2008.
- [Doumeingts 84] G. Doumeingts. *Méthode GRAI : méthode de conception des systèmes en productive*. PhD thesis, Université de Bordeaux 1, 1984.
- [El Haouzi 08a] Hind El Haouzi. *Approche méthodologique pour l’intégration des systèmes contrôlés par le produit dans un environnement de juste-à-temps : Application à l’entreprise TRANE*. Thèse de doctorat, Université Henri Poincaré, Nancy I, France, 2008.
- [El Haouzi 08b] Hind El Haouzi, André Thomas & Jean-François Pétin. *Contribution to reusability and modularity of manufacturing systems simulation models : Application to distributed control simulation within DFT context*. International Journal Production Economics, vol. 112, pages 48–61, 2008.
- [Espinasse 94] B. Espinasse, M. Lai & D. Nanci. *MERISE+ : une extension de la méthode MERISE à l’approche objet par un apport de la méthode HOOD = MERISE+*. In Xe congrès INFORSID, Aix en Provence, France, mai 1994.
- [Evrot 08] Dominique Evrot. *Contribution à la vérification d’exigences de sécurité : Application au domaine de la machine industrielle*. Thèse de doctorat, Université Henri Poincaré, Nancy I, Nancy, France, Août 2008.
- [Féniès 06] Pierre Féniès, Michel Gourgand & Sophie Rodier. *Interoperable and Multi-flow Software Environment : Application to Health Care Supply Chain*. Business Process Management Workshops, pages 311–322, 2006.

- [Fontanili 99] Franck Fontanili. *Intégration d'outils de simulation et d'optimisation pour le pilotage d'une ligne d'assemblage multiproduit à transfert asynchrone*. Thèse de doctorat, Université PARIS 13, 1999.
- [Fontanili 06] F. Fontanili, J. Lamothe, M. Dupuy & C. Nexon. *Etude dynamique de l'ordonancement d'un laboratoire de contrôle qualité pharmaceutique*. In 6e Conférence Francophone de MODélisation et SIMulation, Maroc, 2006.
- [Frankel 03] D.S. Frankel, P. Harmon, J. Mukerji, J. Odell, M. Owen, P. Rivitt, M. Rosen & R.M. Soley. *The Zachman Framework and the OMG's Model Driven Architecture*. Business Process Trends Whitepaper, 2003.
- [Gamma 95] Erich Gamma, Richard Helm, Ralph Johnson & John Vlissides. *Design patterns : elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.
- [Garcia-Ojeda 08] Juan Garcia-Ojeda, Scott DeLoach, Robby, Robby, Walamitien Oyanan & Jorge Valenzuela. *O-MaSE : A Customizable Approach to Developing Multiagent Development Processes*. Agent-Oriented Software Engineering, vol. VIII, pages 1–15, 2008.
- [Gayraud 08] Fabrice Gayraud. *Modélisation et simulation des flux de médicament*. Rapport de stage, IUT de Montluçon, 2008.
- [GERAM 99] GERAM. *GERAM : Generalised Enterprise Reference Architecture and Methodology*. Rapport technique, IFIP-IFAC Task Force on Architectures for Enterprise Integration, 1999.
- [Ghorfi 10] S. Filali El Ghorfi, F. Riane & L. El Hiki. *L'erreur médicamenteuse : vers une analyse rétrospective de l'erreur humaine*. In MOSIM'10 :8e Conférence Internationale de MODélisation et SIMulation, Hammamet, Tunisie, mai 2010.
- [GME 05] GME. *A Generic Modeling Environment : GME 5 User's Manual*. Rapport technique, Institute for Software Integrated Systems Vanderbilt University, 2005.
- [Goujon 97] Jean-Yves Goujon. *Un environnement de modélisation multi-agents pour la spécification et l'évaluation des performances des systèmes industriels de production*. Thèse de doctorat, Université Blaise Pascal - Clermont-Ferrand II, Clermont-Ferrand, France, 1997.
- [Gourgand 91] Michel Gourgand & Patric Kellert. *Conception d'un environnement de modélisation des systèmes de production*. In 3ème congrès international de génie industriel, Tour, France, 1991.
- [Gourgand 92] Michel Gourgand & Patric Kellert. *An object-oriented methodology for manufacturing system modelling*. In Summer Computer Simulation Conference, pages 1123–1128, Reno, Nevada, USA, juillet 1992.
- [Gouyon 04] David Gouyon. *Contrôle par le produit des systèmes d'exécution de la production : apport des techniques de synthèse*. Thèse de doctorat, Université Henri Poincaré, Nancy 1, Nancy, France, décembre 2004.
- [Gouyon 07] David Gouyon, Jean-François Pétin & Gérard Morel. *A product driven re-configurable control for shop floor systems*. Studies in Informatics and Control, vol. 16, 2007.

- [Gouyon 08] David Gouyon & Michael David. *Implantation d'contrôle par le produit à l'aide de réseaux de capteurs sans fil - Validation par émulation d'un environnement de production*. In 7e Conférence Internationale de MODélisation et SIMulation - MOSIM'08, Paris, France, 2008.
- [Gérard 07] S. Gérard, D. Petriu & J. Medina. *MARTE : A New Standard for Modeling and Analysis of Real-Time and Embedded Systems*. Tutorial, 19th Euromicro Conference on Real-Time Systems (ECRTS 07), Pise, Italie, Juillet 2007.
- [Grimaldi 09] Didier Grimaldi. *Modélisation et simulation du circuit du médicament au nouvel hôpital d'estaing*. Rapport de fin d'étude, IFMA, 2009.
- [Guzélian 05] Gwladys Guzélian & Corine Cauvet. *Modèles Sémantiques de Composants pour l'Ingénierie des Systèmes d'Information*. In IC 2005 : 16es journées francophones d'Ingénierie des Connaissances, pages 145–156, juin 2005.
- [Heverhagen 01] T. Heverhagen & R. Tracht. *Negotiation Scenarios between autonomous Robot Cells in Manufacturing Automation : A Case Study*. In Tunisian-German Conference Smart Systems and Devices, pages 499–504, Hammamet, Tunisie, March 2001.
- [Huang 02] C-Y. Huang. *Distributed manufacturing execution systems : A workflow perspective*. Journal of Intelligent Manufacturing, vol. 13, pages 485–497, décembre 2002.
- [Huet 09a] Jean-Charles Huet, Michel Gourgand, Khalid Kouiss & Jean-Luc Paris. *Reengineering methodology for a medication-use process using component-based approach*. In 11th conference on The Modern Information Technology in the Innovation Processes of the Industrial Enterprises, Bergamo, Italie, 2009.
- [Huet 09b] Jean-Charles Huet, Khalid Kouiss, Jean-Luc Paris & Michel Gourgand. *Analogy between medication-use process and assembly system*. In International Conference on Industrial Engineering and Systems Management, Montréal, Canada, 2009.
- [Huet 09c] Jean-Charles Huet, Khalid Kouiss, Jean-Luc Paris & Michel Gourgand. *Conception d'un nouveau système de préparation et de distribution de médicaments dans un hôpital*. In 11ème Colloque National AIP PRIMECA, La Plagne, France, Avril 2009.
- [Huet 10a] Jean-Charles Huet, Jean Chopineau, Vincent Pinon, Isabelle Grémeau, Michel Gourgand, Jean-Luc Paris & Khalid Kouiss. *Proposition d'un outil d'aide à la décision pour l'organisation du circuit du médicament*. In GISEH'10 : Conférence Francophone Gestion et Ingénierie des Systèmes Hospitaliers, Clermont-Ferrand, France, septembre 2010.
- [Huet 10b] Jean-Charles Huet, Khalid Kouiss, Jean-Luc Paris & Michel Gourgand. *Modeling of the interaction between medication-use process and hospital information system*. In ILS 2010 : 3rd International Conference on Information Systems, Logistics and Supply Chain, Casablanca, Maroc, avril 2010.
- [Huet 10c] Jean-Charles Huet, Jean-Luc Paris, Michel Gourgand & Khalid Kouiss. *Modèle de connaissance générique du circuit du médicament dans un hôpital*. In MOSIM'10 : 8ème ENIM IFAC Conférence Internationale de Modélisation et Simulation, Hammamet, Tunisie, mai 2010.

- [IEC60848 88] Norme IEC60848. *Langage de spécification GRAFCET pour diagrammes fonctionnels en séquence*, 1988.
- [IEC61499 05] Norme IEC61499. *Function Blocks*, 2005.
- [IEEE 00] IEEE. The authoritative dictionary of iee standards terms. IEEE Computer Society Press, Los Alamitos, CA, USA, 2000.
- [Ishiwata 97] J. Ishiwata. *Ie for the shop floor : Productivity through process analysis*. Thomson-Shore, Inc, 1997.
- [Kisti 05] M. Kisti, K. Kouiss, F. Chiron, S. Hajri-Gabouj, D.Servat & T. Meurisse. *Composants d'IHM modulaire pour les systèmes de production : Méthodologie et développement*. In 4th International Conference Integrated Design and Production CPI, Casablanca, Maroc, 2005.
- [Kleppe 03] Anneke G. Kleppe, Jos Warmer & Wim Bast. *Mda explained : The model driven architecture : Practice and promise*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.
- [Koestler 67] A. Koestler. *The ghost in the machine*, volume ISBN 0-14-019162-5. 1967.
- [Kosanke 99] K. Kosanke & M. Zelm. *CIMOSA modelling processes*. *Computers in Industry*, vol. 40, no. 2-3, pages 141 – 153, 1999.
- [Kuras 06] Kuras. *A Multi-Scale Definition of a System*. MITRE Technical Paper, vol. [http :// www.mitre.org](http://www.mitre.org) (accessible au 27 octobre 2010), 2006.
- [Laborie 06] Hervé Laborie & Sébastien Woynar. *Organisation du circuit du médicament dans les hôpitaux et cliniques*. Rapport technique, mission nationale d'audit et d'expertise hospitaliers, septembre 2006.
- [Labrosse 06] Hélène Labrosse. *La dispensation nominative journalière automatisée : mise en place dans une pharmacie à usage intérieur*. Thèse de doctorat en pharmacie, université Claude Bernard - Lyon 1, Lyon, France, septembre 2006.
- [Lacomme 98] Philippe Lacomme. *Optimisation des systèmes de production : méthodes stochastiques et approche multiagents*. Thèse de doctorat, Université de Blaise Pascal - Clermont-Ferrand II, 1998.
- [Le Moigne 77] Jean-Louis Le Moigne. *La théorie du système général*, volume 2130433235. 1977.
- [Le Moigne 92] Jean-Louis Le Moigne. *La modélisation des systèmes complexes*. 1992.
- [Leclaire 09] Patrice Leclaire. *Heuristiques et Métaheuristiques pour des problèmes d'optimisation combinatoire : problème d'équilibrage de lignes d'assemblage et problème de tournées de véhicules*. Thèse de doctorat, Université Blaise Pascal - Clermont-Ferrand II, Clermont-Ferrand, France, 2009.
- [Lee 99] Seung-Hyun Lee, James E. Diekmann, Anthony D. Songer & Hyman Brown. *Identifying waste : Applications of construction process analysis*. In Seventh Annual Conference of the International Group for Lean Construction, California, Berkeley, CA, USA, Juillet 1999.
- [Leitão 04] Paulo Leitão. *An Agile and Adaptive Holonic Architecture for Manufacturing Control*. PhD thesis, Dept. Electrical and Computer Eng., University of Porto, Portugal, 2004.

- [Leitão 06] Paulo Leitão & Francisco Restivo. *ADACOR : A holonic architecture for agile and adaptive manufacturing control*. *Computers in Industry*, vol. 57, no. 2, pages 121–130, February 2006.
- [Lohr 02] Christophe Lohr & Jean-Pierre Courtiat. *From the Specification to the Scheduling of Time-Dependent Systems*. In *FTRTFT '02 : Proceedings of the 7th International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems*, pages 129–146, London, UK, 2002. Springer-Verlag.
- [loi 99] *Arrêté du 31 mars 1999 et article R. 5015-48 relatif à la prescription, à la dispensation et à l'administration des médicaments*, 1999.
- [Luzeaux 08] D. Luzeaux & J.R. Ruault. *Ingénierie des systèmes de systèmes - concepts et illustrations pratiques*, volume ISBN 978-2-7462-1875-8. Hermes Science, 2008.
- [Madic 01] Agnès Madic. *Informatisation du circuit du médicament à l'hôpital : de l'intention à la réalisation*. Thèse de doctorat en informatique, université d'Angers, Angers, France, 2001.
- [Majchrzak 06] Yann Majchrzak & Ghislaine Dervieux. *Les nouvelles technologies au service de la sécurisation du circuit du médicament*. In *4th conference GISEH*, pages 183–190, Lauzanne, Suisse, 2006.
- [Marcotte 94] François Marcotte. *Contribution à la modélisation des systèmes de production : extension du modèle GRAI*. Thèse de doctorat, Université de Bordeaux I, Bordeaux, France, 1994.
- [Mařík 02] Vladimír Mařík, Martyn Fletcher & Michal Pěchouček. *Holons & Agents : Recent Developments and Mutual Impacts*. In *Proceedings of the 9th ECCAI-ACAI/EASSS 2001, AEMAS 2001, HoloMAS 2001 on Multi-Agent-Systems and Applications II-Selected Revised Papers*, pages 233–267, London, UK, 2002. Springer-Verlag.
- [Mayer 92] Richard J. Mayer, Michael K. Painter & Paula S. deWitte. *IDEF Family of Methods for Concurrent Engineering and Business Re-engineering Applications*. Knowledge-Based Systems, Inc., 1992.
- [Mcfarlane 00] D. C. Mcfarlane & S. Bussmann. *Developments in holonic production planning and control*. *Production Planning and Control*, vol. 11, no. 6, pages 522–536, September 2000.
- [Mekki 10] A. Mekki, M. Ghazel & A. Toguyeni. *Time-constrained systems validation using MDA model transformation*. In *MOSIM'10 :8ème ENIM IFAC Conférence Internationale de Modélisation et Simulation*, Hammamet, Tunisie, mai 2010.
- [Mertins 97] K. Mertins, R. Jochem & F. W. Jäkel. *A tool for object-oriented modelling and analysis of business processes*. *Computers in Industry*, vol. 33, no. 2-3, pages 345 – 356, 1997. *Co-Operation in Manufacturing Systems, CIM at Work*.
- [Mille 08] Frederic Mille. *Systèmes de détection des interactions médicamenteuses : points faibles et propositions d'améliorations*. Thèse de doctorat, Université Pierre et Marie Curie, Paris, France, décembre 2008.
- [Morel 07] Gérard Morel, Paul Valckenaers, Jean-Marc Faure, Carlos E. Pereira & Christian Diedrich. *Manufacturing plant control challenges and issues*. *Control*

- Engineering Practice, vol. 15, no. 11, pages 1321 – 1331, 2007. Special Issue on Manufacturing Plant Control : Challenges and Issues - 11th IFAC INCOM'04 Symposium on Information Control Problems in Manufacturing.
- [Naumenko 03] Andrey Naumenko & Alain Wegmann. *Two approaches in system modeling and their illustrations with MDA and RM-ODP*. In the 5th International Conference on Enterprise Information Systems, 2003.
- [OMG 03] OMG. Mda guide version 1.0.1, volume omg/2003-06-01. juin 2003.
- [OMG 10a] OMG. Omg systems modeling language (omg sysml) version 1.2, volume formal/2010-06-02. 2010.
- [OMG 10b] OMG. Omg unified modeling language : Superstructure, volume formal/2010-05-06. 2010.
- [Pannequin 07] Rémi Pannequin. *Proposition d'un environnement de modélisation et de test d'architectures de pilotage par le produit de systèmes de production*. Thèse de doctorat, Université Henri Poincaré, Nancy-I, Nancy, France, Juillet 2007.
- [Paris 93] Jean-Luc Paris. *Modélisation et simulation orientées objets pour la conception et l'exploitation d'ateliers flexibles d'assemblage automatique réactifs aux aléas de qualité*. Thèse de doctorat, UFR des sciences et des techniques de l'Université de France-Comté, 1993.
- [Petit 02] Michaël Petit & Guy Doumeingts. *Deliverable D1.1 Report on the State of the Art in Enterprise Modelling*. Thematic Network - Contract IST - 2001 - 34229, University of Namur, septembre 2002.
- [Pierreval 90] Henri Pierreval. *Les méthodes d'analyse et de conception des systèmes de production*, volume ISBN : 2-86601-201-1. 1990.
- [Pélissier 00] E. Pélissier & L. Nguyen. *Traçabilité des produits sanguins labiles : définition, réglementation, bilan et perspectives*. Transfusion Clinique et Biologique, vol. 7, no. Supplement 1, pages 72s – 74s, 2000. XXème Congrès de la Société française de transfusion sanguine.
- [Pétin 07] Jean-François Pétin, David Gouyon & Gérard Morel. *Supervisory synthesis for product-driven automation and its application to a flexible assembly cell*. IFAC Control Engineering Practice journal, vol. 15, Issue 5, no. ISSN 0967-0661, May 2007.
- [Pujo 09] Patrick Pujo, Nicolas Broissin & Fouzia Ounnar. *PROSIS : An isoarchic structure for HMS control*. Engineering Applications of Artificial Intelligence, vol. 22, no. 7, pages 1034 – 1045, 2009. Distributed Control of Production Systems.
- [Rodier 10] Sophie Rodier. *Une tentative d'unification et de résolution des problèmes de modélisation et d'optimisation dans les systèmes hospitaliers. Application au Nouvel Hôpital d'Estaing*. Thèse de doctorat, Université Blaise Pascal - Clermont II, Clermont-Ferrand, France, 2010.
- [Ross 77] D.T. Ross & K.E. Schoman. *Structured Analysis for Requirements Definition*. IEEE Transactions on Software Engineering, vol. 3, pages 6–15, 1977.
- [Roux 10a] O. Roux, D. Duvivier, E. Ramat, G. Quesnel & C. Combes. *UML-Statechart modeling tool for the VLE simulator : an application to a chronic renal dialysis unit*. In MOSIM'10 :8ème ENIM IFAC Conférence Internationale de Modélisation et Simulation, Hammamet, Tunisie, mai 2010.

- [Roux 10b] Olivier Roux. Interopérabilité & circuit du médicament : Le standard pn13 siph2v2. juin 2010.
- [Sallez 04] Y. Sallez, D. Trentesaux, T. Berger & C. Tahon. *Product-based and resource-based heterarchical approaches for dynamic FMS scheduling*. Computers & Industrial Engineering, vol. 46, pages 611–623, 2004.
- [Sarramia 02] David Sarramia. *ASCI mi : une méthodologie de modélisation multiple et incrémentielle. Application aux systèmes de trafic urbain*. Thèse de doctorat, Université Blaise Pascal - Clermont-Ferrand II, Clermont-Ferrand, France, 2002.
- [Scheer 94] August-Wilhelm Scheer. *ARIS Toolset : A software product is born*. Information Systems, vol. 19, no. 8, pages 607 – 624, 1994.
- [Schmitt 06] Étienne Schmitt, Daniel Antier, Chantal Bernheim, Édith Dufay, Marie Caroline Husson & Edgar Tissot. Dictionnaire français de l'erreur médicamenteuse. Numéro ISBN : 2-9526010-0-3. janvier 2006.
- [Servat 05] David Servat, Fabien Chiron, Khalid Kouiss, Thomas Meurisse & Guy Souchet. *Modèle d'architecture générique pour la supervision de systèmes de production*. In Premières Journées sur l'Ingénierie Dirigée par les Modèles IDM'05, Paris, France, 2005.
- [Shack 06] Shack & Tulloch. *The Medical Center at Bowling Green Re-Engineers its Pharmacy with Automation Technologies, Re-Energizes Patient Safety*. 2006.
- [Shen 06] Weiming Shen, Qi Hao, Hyun Joong Yoon & Douglas H. Norrie. *Applications of agent-based systems in intelligent manufacturing : An updated review*. Advanced Engineering Informatics, vol. 20, no. 4, pages 415 – 431, 2006.
- [Silveira 01] Giovanni Da Silveira, Denis Borenstein & Flávio S. Fogliatto. *Mass customization : Literature review and research directions*. International Journal of Production Economics, vol. 72, no. 1, pages 1 – 13, 2001.
- [Simão 05] Jean Marcelo Simão. *Contribution au développement d'un outi de simulation de systèmes holonique de production et proposition d'un méta-modèle de contrôle holonique*. Thèse en co-tutelle, centro federal de educação tecnológica do paraná et Université Henri Poincaré, Paraná, Brésil et Nancy, France, juin 2005.
- [Sowa 92] John F. Sowa & John A. Zachman. *Extending and formalizing the framework for information systems architecture*. IBM Syst. J., vol. 31, no. 3, pages 590–616, 1992.
- [Spry 05] Charles W. Spry & Mark A. Lawley. *Evaluating hospital pharmacy staffing and work scheduling using simulation*. In WSC '05 : Proceedings of the 37th conference on Winter simulation, pages 2256–2263. Winter Simulation Conference, 2005.
- [Starfield 00] Barbara Starfield. *Is US Health Really the best in the world ?* JAMA, vol. 284, no. 4, pages 484–485, juin 2000.
- [Tardieu 83] H. Tardieu, A. Rochfeld & R. Colletti. *La méthode merise - tome 1 principes et outils*, volume ISBN 2-7081-1106-X. 1983.
- [Tolk 03] Dr. Andreas Tolk & James A. Muguira. *The levels of conceptual interoperability model*. In Fall Simulation Interoperability Workshop, Orlando, Florida, September 2003.

- [Trentesaux 02] Damien Trentesaux. *Pilotage hétérarchique des systèmes de production*. Thèse d'hdr, Université de Valenciennes et du Hainaut-Cambrésis (UVHC), 2002.
- [Trentesaux 08] D. Trentesaux, T. Berger & Y. Sallez. *Product & resource driven control of allocation and routing processes in FMS*. In 7e Conférence Internationale de MODélisation et SIMulation, Paris, France, 2008.
- [Ulmer 10] J.S. Ulmer, J.P. Belaud & J.M. Le Lann. *Towards a pivotal-based approach for business process alignment*. In MOSIM'10 :8ème ENIM IFAC Conférence Internationale de Modélisation et Simulation, Hammamet, Tunisie, mai 2010.
- [Valckenaers 98] Paul Valckenaers, Hendrik Van Brussel, Jo Wyns, Luc Bongaerts & Patrick Peeters. *Designing Holonic manufacturing systems*. Robotics and Computer-Integrated Manufacturing, vol. 14, no. 5-6, pages 455 – 464, 1998.
- [Vallespir 03] Bruno Vallespir, Christian Braesch, Vincent Chapurlat & Didier Crestani. *L'intégration en modélisation d'entreprise : Les chemins d'UEML*. In 4e Conférence Francophone de MODélisation et SIMulation, avril 2003.
- [Vernadat 02] F. B. Vernadat. *Enterprise modeling and integration (EMI) : Current status and research perspectives*. Annual Reviews in Control, vol. 26, no. 1, pages 15 – 25, 2002.
- [Viry 10] Julien Viry. Modélisation et simulation du circuit du médicament. Rapport de stage technique, IFMA, 2010.
- [Vogel 08] Laure Vogel. Applicabilité de la technologie rfid au suivi des médicaments au nhe : Réalisation d'un démonstrateur. Rapport de fin d'étude, IFMA, 2008.
- [Williams 94] Theodore J. Williams. *The Purdue enterprise reference architecture*. Computers in Industry, vol. 24, no. 2-3, pages 141 – 158, 1994.
- [Wong 03] Cathy Wong, Glen Geiger, Yaron D. Derman, Carolyn R. Busby & Michael W. Carter. *Healthcare process analysis : redesigning the medication ordering, dispensing, and administration process in an acute care academic health sciences centre*. In WSC '03 : Proceedings of the 35th conference on Winter simulation, pages 1894–1902. Winter Simulation Conference, 2003.
- [Wooldridge 00] Michael Wooldridge, Nicholas R. Jennings & David Kinny. *The Gaia Methodology for Agent-Oriented Analysis and Design*. Autonomous Agents and Multi-Agent Systems, vol. 3, no. 3, pages 285–312, September 2000.
- [Yurtkuran 08] Alkin Yurtkuran & Erdal Emel. *Simulation based decision-making for hospital pharmacy management*. In WSC '08 : Proceedings of the 40th Conference on Winter Simulation, pages 1539–1546. Winter Simulation Conference, 2008.

A

ANNEXES

SOMMAIRE

A.1	LE CIRCUIT PNEUMATIQUE DU CHU ESTAING ET SA REPRÉSENTATION SOUS AUTOMOD	182
A.1.1	Le circuit pneumatique utilisé au CHU Estaing	182
A.1.2	Le synoptique des circuits pneumatiques et leur représentation dans Automod	182
A.2	COMPLÉMENT D'INFORMATION SUR LE SUIVI DU CIRCUIT DU MÉDICAMENT PAR IDENTIFICATION RADIOFRÉQUENCE (RFID)	184
A.2.1	Programmation des tags	184
A.2.2	Exemple de code pour l'écriture d'un tag en C#	185
A.3	EXPLICATION DU CODE AUTOMOD POUR UNE PHARMACIE GÉNÉRIQUE AVEC UNE DISPENSATION NOMINATIVE	186
A.3.1	Création des boîtes correspondant aux patients	186
A.3.2	Choix du parcours de la boîte du patient	186
A.3.3	Arrivée dans le stock du service	188
A.4	DÉTAIL DE L'IMPLANTATION D'UN MODÈLE DANS AUTOMOD	189
A.4.1	Comment implanter un sous modèle	189
A.4.2	Modifications apportées au modèle de l'hôpital Estaing	189

A.1 LE CIRCUIT PNEUMATIQUE DU CHU ESTAING ET SA REPRÉSENTATION SOUS AUTOMOD

A.1.1 Le circuit pneumatique utilisé au CHU Estaing

Le CHU Estaing de Clermont-Ferrand utilise un circuit pneumatique pour l'envoi de certaines doses de médicaments vers les services. Ce circuit pneumatique provient de la société AEROCOM. Les éléments du circuit pneumatique sont : les cartouches (figure A.1a), les stations d'envoi présentes à la pharmacie vers les services (figure A.1b) et des stations de réception au niveau du service (figure A.1c).

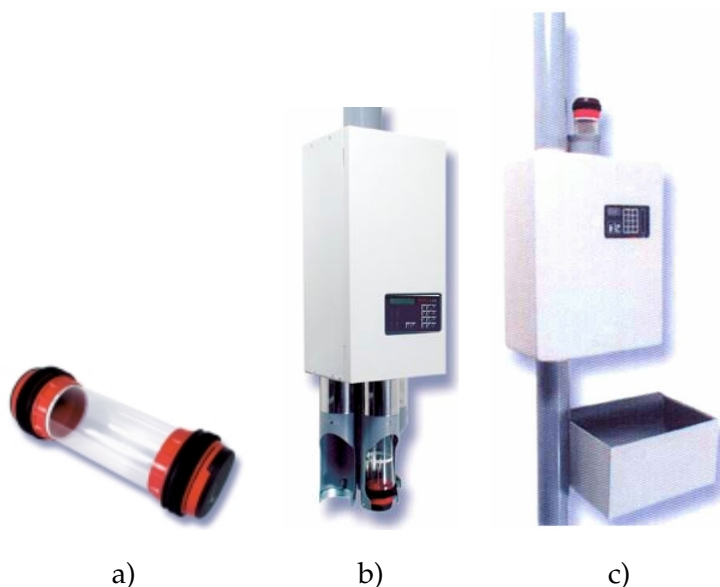


FIGURE A.1 – Cartouche et stations d'envoi et de réception du circuit pneumatique

A.1.2 Le synoptique des circuits pneumatiques et leur représentation dans Automod

Le circuit pneumatique est composé de 3 circuits indépendants qui gèrent chacun une partie du CHU Estaing. Les 3 circuits sont donnés sous forme de synoptique comme celui montré sur la figure A.2.

Dans le logiciel de simulation Automod, il est représenté par un *path mover system* qui permet le transport de *vehicules* de façon automatique. Ces *vehicules* symbolisent les cartouches. L'exemple du circuit 1 présenté sur la figure A.2 correspond dans Automod à la figure A.3.

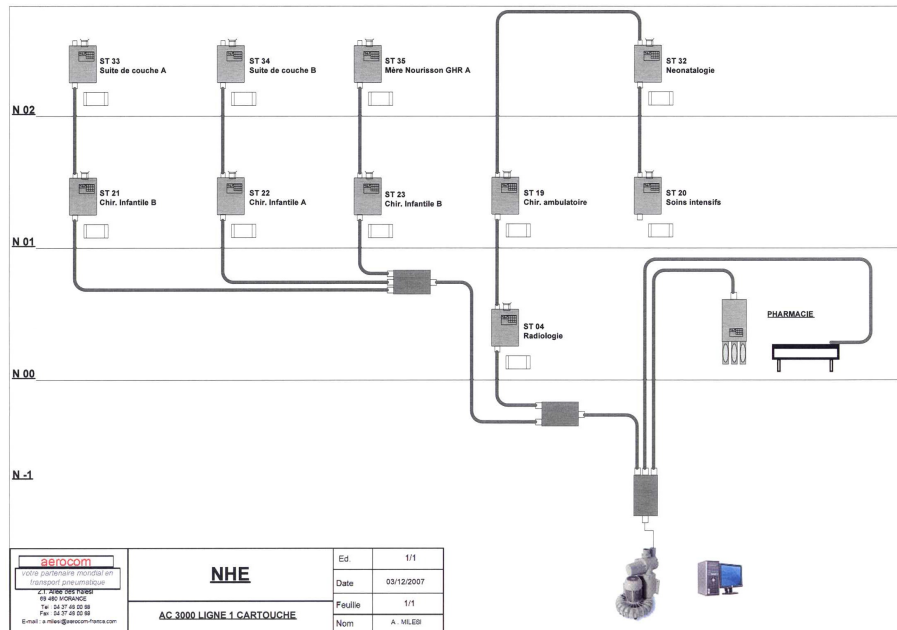


FIGURE A.2 – Synoptique représentant un des trois circuits pneumatiques au CHU Estaing de Clermont-Ferrand

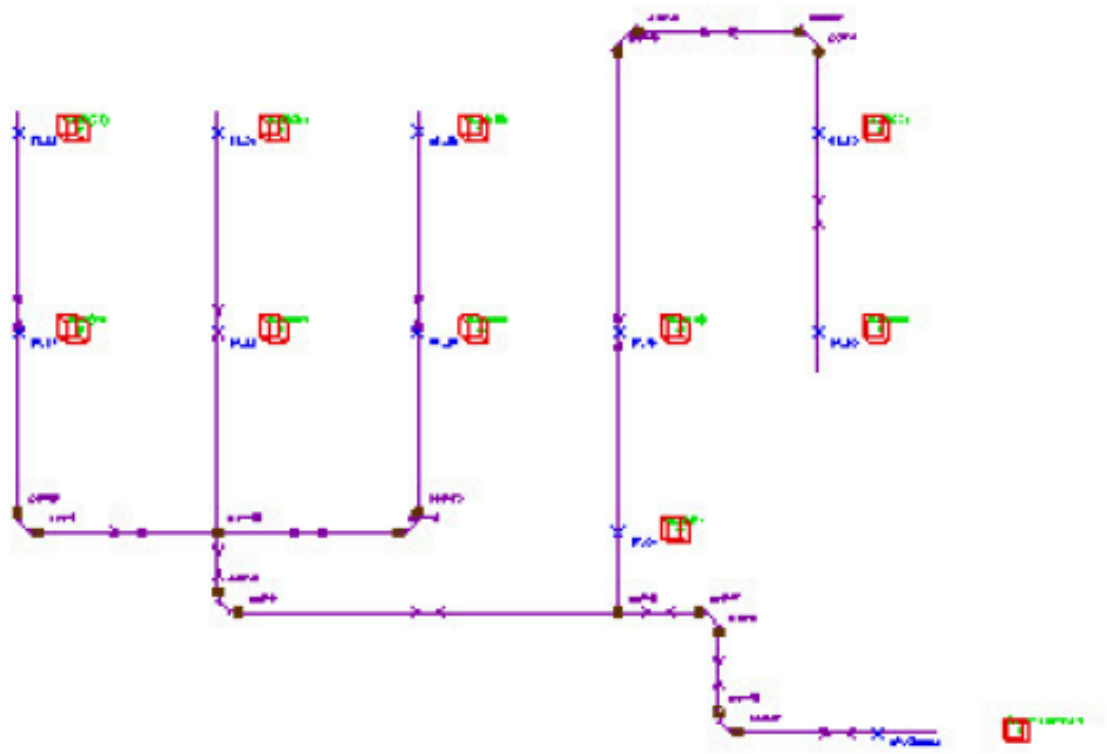


FIGURE A.3 – Modélisation dans Automod du circuit pneumatique

A.2 COMPLÉMENT D'INFORMATION SUR LE SUIVI DU CIRCUIT DU MÉDICAMENT PAR IDENTIFICATION RADIOFRÉQUENCE (RFID)

A.2.1 Programmation des tags

Les tags utilisés ont une capacité de 108 octets (27×4) soit 108 caractères. Chaque ligne est appelée bloc. Chaque bloc est paramétré indépendamment dans une dll, ainsi il suffit d'utiliser les noms des blocs tels qu'ils ont été programmés dans un fichier au format *dll* pour lire et écrire depuis le code C#. La structure de la mémoire des tags utilisés est montrée figure A.4.

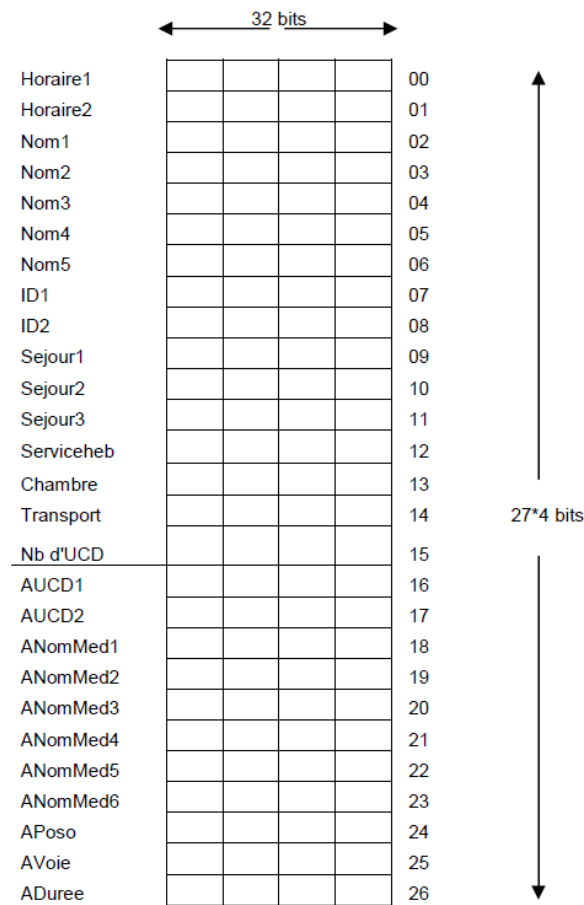


FIGURE A.4 – structure de la mémoire des tags utilisés

A.2.2 Exemple de code pour l'écriture d'un tag en C#

Nous montrons sur la figure A.5 un exemple de code qui agit sur le tag. Il représente ici la vérification au lit du malade. La vérification consiste à comparer directement les données lues sur le tag après conversion en *string*, aux données de la base de données par l'intermédiaire du *datagrid*. Si les données ne correspondent pas, une boîte de dialogue apparaît et renseigne l'utilisateur sur l'endroit où l'écriture ne s'est pas faite correctement.

```
// Lecture ID
int readint7 = Convert.ToInt32(t.ID1); string readhex7 =
readint7.ToString("X");
int readint8 = Convert.ToInt32(t.ID2); string readhex8 =
readint8.ToString("X");
string readID = ConvertHexStringToAsciiString(readhex7 + readhex8);

// Verification ID
object o2 = PreparationDataGrid.Rows[a].Cells[1].Value;
string s2 = Convert.ToString(o2);
if (s2==readID)
{
    id = true;
}
else
{
    MessageBox.Show("ERREUR ECRITURE TAG: ID");
}
.
.
.
if (horaire && nompat && id && sejour && service && chambre && ucd &&
produit && poso && voie && duree)
{
    VERIFbutton.Text = "VERIF OK";
}
```

FIGURE A.5 – Extrait du code pour la vérification de la concordance au lit du malade

A.3 EXPLICATION DU CODE AUTOMOD POUR UNE PHARMACIE GÉNÉRIQUE AVEC UNE DISPENSATION NOMINATIVE

Cet annexe nous permet de préciser quelques points de la programmation du modèle.

A.3.1 Création des boîtes correspondant aux patients

Au temps 0, une ressource est créée une et une seule fois. Elle entre dans la procédure P_Lecture. On commence par ouvrir le fichier *Temps.txt* où sont enregistrés les temps de picking des postes et les temps que mettent les boîtes à entrer et à sortir du convoyeur. Ces données sont enregistrées dans des variables internes d'Automod (figure A.6).

```
open "Temps.txt" for reading save result as V_FilePtrordo0
read V_HeaderArrayordo0 as 100 from V_FilePtrordo0
set V_Ordo to 0
while V_FilePtrordo0 eof = false do
  begin
    inc V_Ordo by 1
    /*lecture de la ligne courante de Temps.txt*/
    read V_Temps(V_Ordo), V_Valeur(V_Ordo) from V_FilePtrordo0
    with delimiter "\t"
  end
```

FIGURE A.6 – Code Automod pour la création d'une boîte qui contiendra les DPP

On lit le fichier *Ordo.txt* qui contient les données des ordonnances. On crée alors une boucle qui pour chaque ligne (correspondant à un médicament d'un patient) l'enregistre (quantité du médicament et numéro du médicament) et passe à la ligne suivante. Si celle-ci correspond toujours au même patient, alors le second médicament est enregistré ; sinon la ressource est clonée et une boîte contenant les informations du patient est créée (figure A.7).

A.3.2 Choix du parcours de la boîte du patient

Les fichiers où sont mis tous les numéros des médicaments disponibles à chaque poste sont lus (*Liste1.txt, Liste2.txt, ... ListeX.txt*). Le but est de comparer les numéros des médicaments enregistrés pour chaque patient aux listes afin de trouver les postes où se rendre. Quand ceux-ci sont trouvés on enregistre le numéro du poste et le nombre de médicaments à prendre dans des attributs de la boîte (figure A.8).

```

open "Ordo.txt" for reading save result as V_FilePtrordo
read V_HeaderArrayordo as 200 from V_FilePtrordo
set V_Ordo to 0
set V_nb_element to 0
while V_FilePtrordo eof = false do
  begin
    inc V_Ordo by 1
    /* lecture de la ligne courante de Ordo.txt*/
    read V_SERVICEP0(V_Ordo), V_SECLEUNIK(V_Ordo), V_PRCLEUNIK(V_Ordo), V_POSOLOGIE1(V_Ordo), V_POSOLOGIE2(V_Ordo)
    with delimiter "\t"
    set V_nb_element=V_Ordo
  end
  set V_Ordo to 0
  while V_Ordo <> V_nb_element do
    begin
      set i to 1
      while i <> 21 do
        begin
          set A_Medoc(i) to 0
          set A_Qt(i) to 0
          inc i by 1
        end
        inc V_Ordo by 1
        /* enregistre le numéro du patient et son UF*/
        set A_Patient=V_SECLEUNIK(V_Ordo)
        set A_UF=V_SERVICEP0(V_Ordo)
        /* enregistre le numéro du 1er médicament et sa quantité*/
        set A_Medoc(1)=V_PRCLEUNIK(V_Ordo)
        set A_Qt(1)=(V_POSOLOGIE1(V_Ordo)+V_POSOLOGIE2(V_Ordo)+V_POSOLOGIE3(V_Ordo)+V_POSOLOGIE4(V_Ordo)+V_POSOLOGIE
        /* passe à la ligne suivante*/
        inc V_Ordo by 1
        set i to 1
        /* regarde si c'est toujours le même patient*/
        while V_SECLEUNIK(V_Ordo)=A_Patient and i<=20 do
          begin
            inc i by 1
            /*si oui, on enregistre le deuxième médicament, et on recommence le test sur la ligne suivante*/
            set A_Medoc(i)=V_PRCLEUNIK(V_Ordo)
            set A_Qt(i)=(V_POSOLOGIE1(V_Ordo)+V_POSOLOGIE2(V_Ordo)+V_POSOLOGIE3(V_Ordo)+V_POSOLOGIE4(V_Ordo)+V_P
            /*passe à la ligne suivante*/
            inc V_Ordo by 1
          end
        if V_SECLEUNIK(V_Ordo) <> A_Patient and V_Ordo <> V_nb_element then
          begin
            /* si non, on met tout les autres médicaments à 0 et on repasse à la 1ère ligne*/
            decrement V_Ordo by 1
          end
        end
      end
    end
  end
end

```

FIGURE A.7 – Code Automod pour la lecture des ordonnances

```

begin P_ChoixParcours arriving procedure
/* on met à 0 les attributs correspondant aux nombres de médicament à prendre à chaque poste*/
set A_Robot1 to 0
set A_Robot2 to 0
set A_Preparateur1 to 0
set A_Preparateur2 to 0
/* on ouvre le fichier (Liste1.txt) correspondant à la liste des médicaments disponibles au poste 1(robot1)*/
open "Liste1.txt" for reading save result as V_FilePtrordo1(procindex)
read V_HeaderArrayordo1(procindex) as 100 from V_FilePtrordo1(procindex)
set V_Ordo1(procindex) to 0
while V_FilePtrordo1(procindex) eof = false do
  begin
    /*on augmente la variable correspondant au numéro de la ligne lu*/
    inc V_Ordo1(procindex) by 1
    /* lit le numéro du médicament dans le fichier texte*/
    read V_NUMMEDOC1(V_Ordo1(procindex),procindex) from V_FilePtrordo1(procindex)
    with delimiter "\t"
    set A_i to 1
    while A_i <= 20 do
      begin
        if V_NUMMEDOC1(V_Ordo1(procindex),procindex)=A_Medoc(A_i) then
          begin
            /*si oui, met le nombre d'exemplaire de ce médicament dans
            l'attribut robot1, pour aller vers ce poste*/
            inc A_Robot1 by A_Qt(A_i)
            /* et met le nombre du médicament à 0, pour être sur de ne pas donner deux fois
            la dose si plusieurs postes ont le même médicament*/
            set A_Medoc(A_i) to 0
          end
          inc A_i by 1
        end
      end
    end
  end
end

```

FIGURE A.8 – Code Automod pour le choix du parcours lors de l'assemblage des DPP

A.3.3 Arrivée dans le stock du service

Le but est de faciliter l'interface avec le modèle de simulation d'un hôpital. Il faut donc avoir le moins possible de données à transmettre. Ici, les médicaments d'un patient se retrouvent dans une file unique, c'est le numéro de cette file qui donne toutes les informations nécessaires sans l'utilisation d'un attribut. En effet, le numéro de Q_Pole indique le numéro du pôle, le service grâce au sous-conteneur UF*x*, et les trois dimensions correspondent à une position du patient dans le stock du service. Les médicaments sont alors prêts à être dispensés.

Le motif présenté à la figure A.9 est répété autant de fois qu'il y a de services et de pôles.

```
begin P_ArriveStockUF arriving procedure
/*on met les médicaments dans la case qui correspond au patient dans le stock de son UF dans son pôle*/
/*on regarde en premier le numéro de l'UF dans le pôle*/
if A_UF=1 then
begin
/*on le met dans le bon pole dont le numéro est dans procindex*/
move into Q_Pole(procindex) container UF1(A_Dim1,A_Dim2,A_Dim3)
end
```

FIGURE A.9 – Code Automod pour le rangement des DPP dans le stock des pôles et des services.

A.4 DÉTAIL DE L'IMPLANTATION D'UN MODÈLE DANS AUTOMOD

A.4.1 Comment implanter un sous modèle

Quand on est dans un modèle, on peut facilement en insérer un autre qui sera alors considéré comme un sous-modèle. Pour cela, il suffit de faire *File -> Insert* et de choisir un fichier au format *.mod* correspondant au modèle que l'on souhaite insérer. Dans notre cas, il y a donc plusieurs possibilités : soit mettre la pharmacie générique en modèle et l'hôpital en sous-modèle, soit l'inverse, soit créer un modèle vide et créer deux sous-modèles. L'intérêt du modèle souche est que ces éléments peuvent être appelés sans connaître le nom précis du modèle en écrivant *modelsys.elementdumodèle*. Comme notre but était d'appeler les médicaments de la pharmacie générique dans l'hôpital, il a été plus judicieux de mettre l'hôpital en sous-modèle de la pharmacie générique.

On obtient alors l'arbre de modèle de la figure A.10 :

```

-pharmacie generique Model
  CircuitBoites      Conveyor
  -modelfinal        Submodel
    Pneu3            Path Mover
    pneu2            Path Mover
    Voie             Path Mover
    pneu1            Path Mover

```

FIGURE A.10 – Arbre des modèles sous Automod

Nous pouvons voir en haut la pharmacie générique et, en dessous, l'hôpital(*modelfinal*) en sous-modèle avec ces constituants (circuit chariot *-voie-* et les trois circuits pneumatiques *-pneu 1, 2, 3-*).

A.4.2 Modifications apportées au modèle de l'hôpital Estaing

Le problème est que lorsqu'il n'y avait qu'une pharmacie (globale), il n'y avait pas besoin de se poser la question de la provenance des médicaments pour un service. Par conséquent, lorsqu'on regarde si des médicaments sont disponibles pour un service, il faut regarder dans les deux pharmacies.

Ainsi,

```

if OL_Chariot(V_UFtourneel(V_t1)) current loads <> 0 then
  devient
if OL_Chariot(V_UFtourneel(V_t1)) current loads <> 0 or
modelsys.OL_Dispensation(V_UFtourneel(V_t1))current loads <> 0 then

```

On ajoute un attribut afin d'enregistrer cette provenance (nommé *A_Who* avec *0* pour la pharmacie globale et *1* pour la pharmacie nominative). De plus, s'il y a une erreur dans les données et qu'il y a des prescriptions en nominatif et en global pour un même service, cet attribut permet de n'utiliser qu'un type de données car il vaut soit *1* soit *0* et rien d'autre. Dans notre cas, si les deux types de données existent, ce sont les données globales qui sont prises en compte, moins lourdes à traiter en cas d'une simulation inutile.

Il faut alors modifier la copie et la suppression des médicaments suivant le type de pharmacie.

```
    set r1 = OL_Chariot(V_UFtournee1(V_t1)) total loads

wait for r1*15 sec

order all loads from OL_Chariot(V_UFtournee1(V_t1))
to P_RemplirUF(V_UFtournee1(V_t1 ))

    devient

if A_Qui=0 then

begin

set r1 = OL_Chariot(V_UFtournee1(V_t1)) total loads

wait for r1*15 sec

order all loads from OL_Chariot(V_UFtournee1(V_t1))
to P_RemplirUF(V_UFtournee1(V_t1 ))

end

else if A_Qui=1 then

begin

set r1 = modelsys.OL_Dispensation(V_UFtournee1(V_t1)) total loads

wait for r1*15 sec

order all loads from modelsys.OL_Dispensation(V_UFtournee1(V_t1))
to P_RemplirUF(V_UFtournee1(V_t1 ))

end
```

Il y a donc simplement l'ajout d'une condition suivant la valeur de l'attribut A_Qui. Avec ces modifications, le modèle traite à la fois la pharmacie nominative et globale, il permet donc maintenant de comparer les deux types de fonctionnement.

De plus, ces modifications sont peu nombreuses, et valident donc la nomination « pharmacie générique » qui peut être implantée dans n'importe quelle simulation d'hôpital existante, ou alors grandement faciliter la réalisation d'une simulation d'un hôpital puisque seul l'existant est à faire, les améliorations de la pharmacie nominative étant déjà prêtes.

Titre Proposition d'une méthodologie de réingénierie pour le contrôle par le produit de systèmes manufacturiers : Application au circuit du médicament d'un hôpital

Résumé Le circuit du médicament est un élément primordial dans l'organisation des établissements de santé. Sa sécurisation et sa modernisation permettent de baisser sensiblement le nombre de personnes victimes d'erreurs médicamenteuses et de réduire les coûts (augmentation du rendement, libération de poste préparateur et gain de temps infirmière...). L'utilisation de la DJIN (Dispensation Journalière Individuelle Nominative), de systèmes de préparation et de transport sécurisés (robots, circuit pneumatique, chariots de dispensation...), de systèmes de contrôle de l'administration au lit du patient sont les solutions permettant d'éviter les erreurs humaines. Nous proposons dans cette étude d'utiliser des concepts du milieu industriel pour améliorer la sécurité et la traçabilité du circuit du médicament. Nous utilisons le paradigme de Système Contrôlé par le Produit (SCP) qui propose de mettre le produit au cœur du système en utilisant des nouvelles technologies tel que la RFID. Nous comparons le circuit du médicament avec un système d'assemblage : l'opération d'assemblage est vue comme un ajout de médicaments et le produit comme la préparation de la prescription pour un patient. Le but de cette thèse est de proposer une méthodologie de réingénierie, basée sur le MDA et ASCI, pour le contrôle par le produit en synchronisant les flux physiques et informationnels, puis de l'appliquer au circuit du médicament. Enfin, nous l'instancions au CHU de Clermont-Ferrand en testant différents scénarios à travers un logiciel de simulation afin de proposer une aide à la décision pour les pharmaciens et les responsables du CHU.

Mots-clés MDA, ASCI, circuit du médicament, Système Contrôlé par le Produit.

Title Proposition of reengineering methodology for the product driven of manufacturing systems : Application to the medication-use process

Abstract The reengineering of the medication-use process is an important challenge in order to reduce medication errors and costs for the pharmacy (increase of the efficiency ; reduce the amount of work time spent by nurses in the medication administration process...). The use of the nominative dispensing, automatic systems (robots, pneumatic circuit, cart dispensing...), control system of administration to the patient are solutions which can avoid the human errors. We propose in this study to use the concept used in industrial systems in order to improve the safety and the traceability on the medication-use process. We use the paradigm of Product Driven System which proposes to place the product in the centre of the system using identification technology as RFID. We made an analogy between the medication-use process and an assembly system : the product is considered as a preparation of the prescription for a patient and an operation of assembly is considered as an addition of medication. The goal of this thesis is to propose a reengineering methodology based on MDA (Model Driven Architecture) and ASDI (Analysis-Specification-Design-Implementation) in order to implement a system based on product driven synchronizing the physical and informational flow. In particular, this methodology is applied to the medication-use process. Finally, we present an example of application of this modelling to the hospital of Clermont-Ferrand through simulation software which allows testing different scenarios to help the pharmacists and managers of the hospital to make up their decisions.

Keywords MDA, ASDI, medication-use process, Product Driven System.

